

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> OMB NO. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comment regarding this burden estimates or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 1998		3. REPORT TYPE AND DATES COVERED Final Report
4. TITLE AND SUBTITLE Design for Packageability. Early Consideration of Packaging From a Microelectronic System Designer's Viewpoint			5. FUNDING NUMBERS DAAH04-94-G-0004	
6. AUTHOR(S) Donald W. Bouldin				
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES) University of Tennessee Knoxville, TN 37996-2100			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211			10. SPONSORING / MONITORING AGENCY REPORT NUMBER ARO 32575.1-EL	
11. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12 b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The goal of this project was to assist in globally optimizing the design of higher performing microelectronic systems by incorporating critical packaging factors into the IC design flow. More specifically, our work was intended to empower system designers, especially designers of multi-million transistor ICs, to obtain optimized MCM-based systems.				
14. SUBJECT TERMS			15. NUMBER OF PAGES	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OR REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	



University of Tennessee Microelectronic Systems Research Laboratory

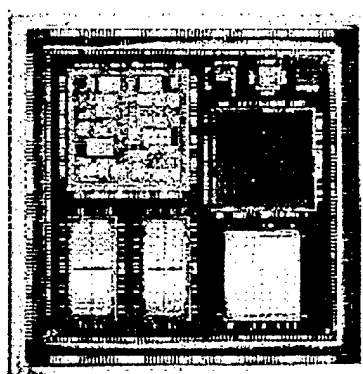
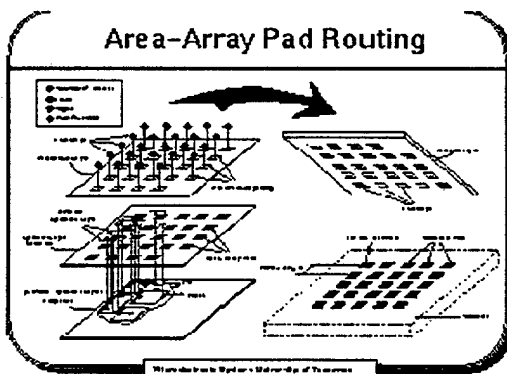
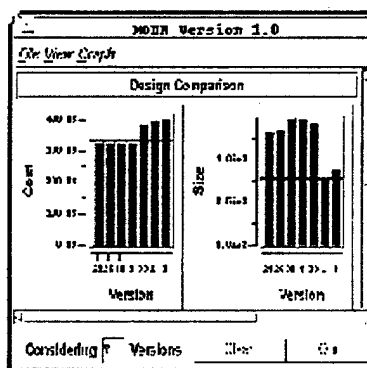
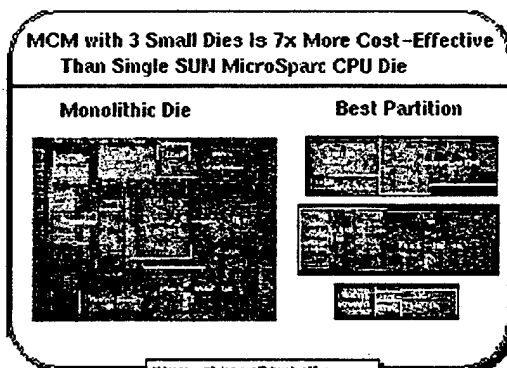


DESIGN FOR PACKAGEABILITY

A research project sponsored by the Defense Advanced Research Projects Agency (DARPA)

PROJECT GOAL: Assist the microelectronic system designer in exploring the design space in order to perform an optimization that takes full advantage of MCM packaging.

For more information, click on one of the following:



PRINCIPAL INVESTIGATOR: Don Bouldin, Ph.D.

Electrical Engineering
1508 Middle Drive
University of Tennessee
Knoxville, TN 37996-2100

TEL: (423)-974-5444
FAX: (423)-974-8245
Email: dbouldin@utk.edu
WWW: <http://www.ee.utk.edu/>

http://microsys6.engr.utk.edu/ece/research/microsys/arpa_project.html

DTIC QUALITY INSPECTED 2

19980519 174

DESIGN FOR PACKAGEABILITY

Early Consideration of Packaging

from a Microelectronic System Designer's Viewpoint

DARPA Order A879

DHHS A-04-94-04532

AMXRO-AAA: P-32575-EL

FINAL REPORT for 1 Jan 94 - 31 Dec 97

by

Don Bouldin, Ph.D.

Principal Investigator

Electrical Engineering

1508 Middle Drive

University of Tennessee

Knoxville, TN 37996-2100

TEL: (423)-974-5444

FAX: (423)-974-8245

EMAIL: dbouldin@utk.edu

EXECUTIVE SUMMARY

The goal of this project was to assist in globally optimizing the design of higher performing microelectronic systems by incorporating critical packaging factors into the IC design flow. More specifically, our work was intended to empower system designers, especially designers of multi-million transistor ICs, to obtain optimized MCM-based systems.

This work is innovative with respect to the following:

- The concept of Design for Packageability had not been fully developed nor documented.
- The procedure for using area-array pads rather than peripheral pads had not been documented and promulgated publicly.
- A comprehensive system model that can be used to analyze and compare from a system designer's viewpoint the impact of different packaging technologies and different IC design methodologies did not exist.
- Early analysis CAD software incorporating such a system model was not available.
- Commercial CAD software for designing ICs did not include consideration of critical packaging parameters.
- MCM prototyping via a multi-project broker had not been conducted by university designers.

Multi-chip integration technology has the potential to increase significantly overall system performance and reliability while reducing size, weight, power and cost. The work conducted for this project explored and demonstrated the impact of these MCM-based technologies on IC design and, hence, on the overall system. Development and verification of the Design for Packageability procedures was performed to accelerate the architectural exploitation of both ICs and MCMs.

Thus, the major accomplishments of this effort are:

- **Early Analysis and Design Space Exploration Tools**

We developed a set of CAD tools which permit exploration of the IC and packaging design space simultaneously. This approach emphasizes concurrent consideration of the partitioning of a microelectronic circuit design into multiple dies and the selection of the appropriate packaging technology for implementation of the entire system. Partitioning a large design into a multichip package is a non-trivial task. Similarly, selection of the MCM packaging technology to accommodate a multichip solution can also be puzzling. The interdependencies of these two problems afford the opportunity to achieve a global optimum when considered concurrently. In our tool we address the partitioning/MCM technology tradeoff and their interdependency. The SUN MicroSparc CPU was used as a demonstration vehicle and was partitioned for different MCM technologies. The results show that the optimum number of partitions and contents of each partition depend heavily on the choice of MCM technologies for a given application.

- **Area-array pad router**

Arranging I/O in a matrix array over the core circuitry of an IC generally provides 5- 10 times more I/O than the traditional method of restricting pads to the periphery. This approach also minimizes overall die size. This method was pioneered by IBM over thirty years ago and has recently become attractive for new designs requiring several hundred I/O. In this project we developed a new area-array pad router which differs from other approaches in that no additional metal layer is added (unless needed) and no redistribution is required. We developed a design guideline definition, data preparation, pad placement, pad assignment, pad routing, and output padframe generation for this technique. The new router was used to provide for area-array I/O on the partitioned MCM dies requiring 112, 298 and 485 I/Os.

- **MCM Prototyping Via MIDAS**

Our project was the one of the first to exercise the new multi-project prototyping service introduced by MIDAS. We designed a general-purpose programmable DSP subsystem that was packaged in a multichip module. The subsystem contained a 32-bit floating-point programmable DSP processor along with 256 K-bytes of SRAM, 128 K-bytes of EEPROM, a 10 K-gate FPGA and a 6-channel 12-bit ADC. The complete subsystem was interconnected on a 37 mm by 37 mm MCM-D substrate and packaged in a 320-pin ceramic quad flat pack. The design was submitted to the MIDAS brokerage service and fabricated by Micro Module Systems. Our experience showed that low-volume MCM prototyping is achievable and somewhat affordable for universities. The design flow, electrical and thermal analyses, CAD tools, cost and lessons learned were documented.

IMPACT AND TECHNOLOGY TRANSFER

This research project has made significant impact on this emerging field in the following ways:

1. Our early analysis approach which permits MCM-based system optimization by incorporating critical packaging factors into the design of the integrated circuits is now being adopted by other designers. Specifically, the title and scope of the IEEE Multi-Chip Module Conference which has been held annually in Santa Cruz, CA, for the past seven years was changed in 1998 to IEEE Symposium on IC/Package Design Integration. Several papers from industry and academia were presented using the approach which we pioneered.
2. The early analysis tools that we developed are available for others to use. Following a presentation we made at Georgia Tech, Prof. Sudha Yalamanchili visited us and has since adopted our approach.
3. The area-array pad router for ICs is now available for others to use. Our work stimulated and complemented the development of a similar commercial tool developed by Cascade Design Automation.
4. Our pioneering the design and prototyping of a MCM via the MIDAS brokerage has enabled MIDAS to adopt regularly scheduled fabrication runs. Our experiences resulted in the principal graduate student designer winning the First Place Award in an international contest sponsored by the Mentor Graphics Corporation. Furthermore, the procedure for MCM design and prototyping is being used in a graduate course at the University of Tennessee to serve as a model for other universities to follow.
5. Dr. Peyman Dehkordi, one of the principal contributors to this DARPA project, has launched a commercial spin-off company to assist industry and government agencies in exploiting the fruits of this research. The company, called Applied Computing Technologies, is located in nearby Oak Ridge, TN. It anticipates designing a MCM for the Air Force Research Lab in Albuquerque as part of

their Highly Integrated Packaging and Processing program. Furthermore, a new commercial MCM product is being developed by the company.

6. Another form of technology transfer is the production of newly trained graduates who can help industry in a variety of ways. Individuals who were involved in this research project and their current employers include:

Researcher	Employer
Chattapadhyay, Subho	Motorola, Austin, TX
Dehkordi, Peyman	ACT, Oak Ridge, TN
Devine, Quinn	Intel, Phoenix, AZ
Ghazi, Sassine	Intel, Phoenix, AZ
Govindan, Rajeev	QualComm, San Diego, CA
Neiroukh, Osama	Intel, Hillsboro, OR
Powell, Tim	Cadence, Cary, NC
Ramamurthi, Karthi	Intel, Sacramento, CA
Shen, Zijun	Analog Devices, Greenboro, NC
Tan, Chandra	Univ. of Tennessee, Knoxville, TN
Tolnas, Barry	Intel, DuPont, WA
Williams, Nicolas	Tanner EDA, Pasadena, CA
York, John	Lockheed 3-D, Orlando, FL

APPENDICIES

The following publications and presentations document in detail the work conducted for this project. Since these papers were peer-reviewed, their acceptance by the community certifies the significance and quality of our efforts.

1. Bouldin, D., "Design for Packageability", Presentation at DARPA Principal Investigators' Meeting, May 1997.
2. Dehkordi, P., Ramamurthi, K., Bouldin, D., Davidson, H., and P. Sandborn, "Impact of MCM Technology on Microelectronic Systems Partitioning: A Case Study", *Proceedings of 1995 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, February 1-2, 1995.
3. Bouldin, D. and P. Dehkordi, "Design for Packageability: An Overview", *Proceedings of 1995 International Intersociety Electronic Packaging Conference*, Lahaina, HI, March 26-30, 1995.
4. Chattapadhyay, S., Dehkordi, P., and D. Bouldin, "An Overview of Placement and Routing Algorithms for Multi-Chip Modules", *International Journal of High Speed Electronics - Special Issue on CAD for Multichip Modules*, December, 1995.
5. Dehkordi, P., Powell, T., and D. Bouldin, "Development of a DSP/MCM Subsystem: Assessing Low-Volume, Low-Cost MCM Prototyping for Universities", *Proceedings of 1996 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, February 5-7, 1996.
6. Dehkordi, P., Ramamurthi, K., Bouldin, D., and H. Davidson, "Early Cost/Performance Cache Analysis of a Split MCM-Based MicroSparc CPU", *Proceedings of 1996 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, February 5-7, 1996.
7. Dehkordi, P., Powell, T., and D. Bouldin, "Performance Comparison of MCM-D and SMT Packaging Technologies for a DSP Subsystem", *Proceedings of 1996*

IEEE International Symposium on Circuits and Systems Conference, pp. IV-245 thru IV-248, Atlanta, GA, May 12-15, 1996.

8. Dehkordi, P., Chandra, T. and D. Bouldin, "Intrinsic Area Array ICs: What, Why, and How", *Proceedings of 1997 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, February 4-5, 1997.
9. Dehkordi, P., Ramamurthi, K., Bouldin, D. and H. Davidson, "Determination of Optimum Area-Array Bond Pitch for MCM Systems", *Proceedings of 1997 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, February 4-5, 1997.
10. Tan, C., Bouldin, D. and P. Dehkordi, "An Intrinsic Area-Array Pad Router for ICs", *Proceedings of Tenth Annual IEEE International ASIC Conference*, pp. 265-269, Portland, OR, September 7-10, 1997.
11. Tan, C., Bouldin, D. and P. Dehkordi, "Design Implementation of Intrinsic Area Array ICs", *Proceedings of 1997 Advanced Research in Very Large Scale Integration Conference*, Ann Arbor, MI, September 15-16, 1997.
12. York, J., Powell, T., Dehkordi, P. and D. Bouldin, "Enhancement of MCM Testability Using an Embedded Reconfigurable FPGA", *Proceedings of 1997 International Conference on Innovative Systems in Silicon*, pp. 165-173, Austin, TX, October 8-10, 1997.

DESIGN FOR PACKAGEABILITY
EARLY CONSIDERATION OF PACKAGING FROM A
MICROELECTRONIC SYSTEM DESIGNER'S VIEWPOINT

University of Tennessee

<i>Dr. Don Bouldin</i>	<i>Dr. Peyman Dehkordi</i>
<i>Subho Chattopadhyay</i>	<i>Quinn Devine</i>
<i>Raj Kanagara</i>	<i>Danny Newport</i>
<i>Tim Powell</i>	<i>Karthi Ramamurthi</i>
<i>Chandra Tan</i>	<i>Barry Tolnas</i>

SUN	MCC
<i>Dr. Howard Davidson</i>	<i>Dr. Peter Sandborn</i>

DARPA

Bob Parker ITO/ETO

bouldin@microsys1.engr.utk.edu
<http://www-ece.engr.utk.edu/ece/research/microsys/microsys.html>
TEL: (423)-974-5444 FAX: (423)-974-8245

Microelectronic Systems--University of Tennessee

DESIGN FOR PACKAGEABILITY:
GOAL AND APPROACH

GOAL:

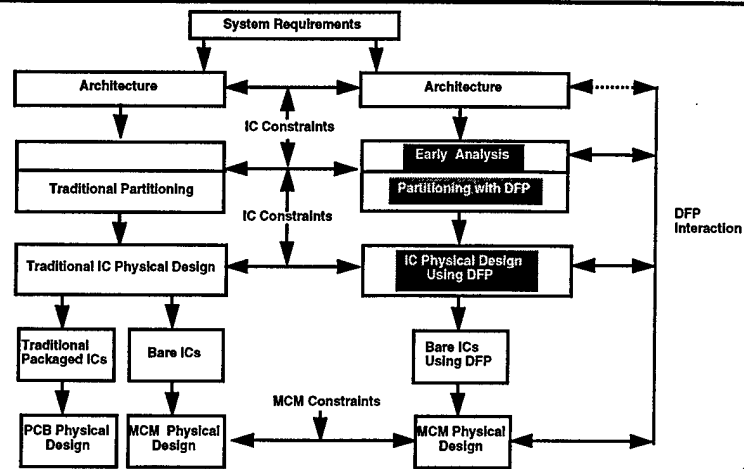
Assist the microelectronic system designer in exploring the design space in order to perform an optimization that takes full advantage of MCM packaging.

APPROACH:

Develop a Design-for-Packageability tool to explore IC partitioning and MCM options at an early stage of the design. Coordinate IC and MCM physical design for global optimization.

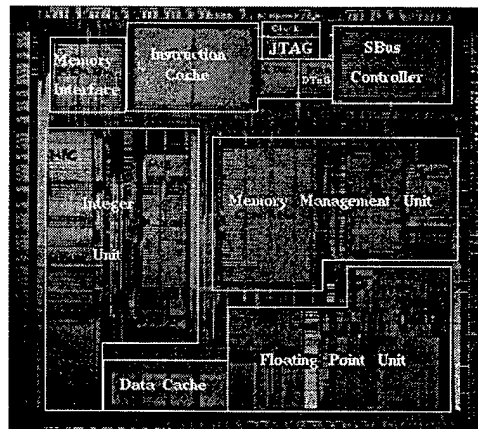
Microelectronic Systems--University of Tennessee

System Design Should Include Design For Packageability



Microelectronic Systems--University of Tennessee

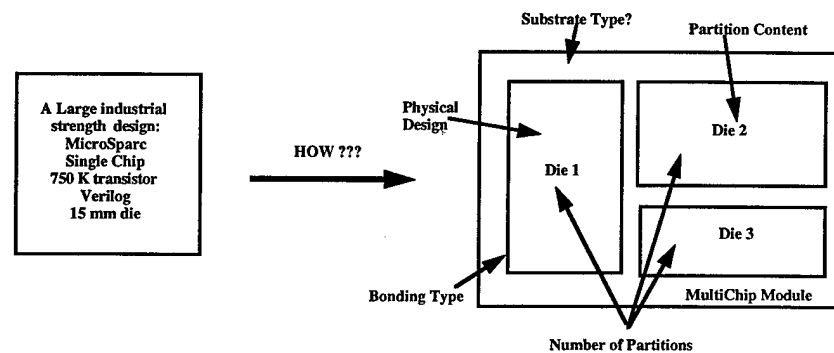
Problem: Partition a Large Design into Multiple Dies for a Selected Multichip Module



Microelectronic Systems--University of Tennessee

Design for Packageability

MOTIVATION



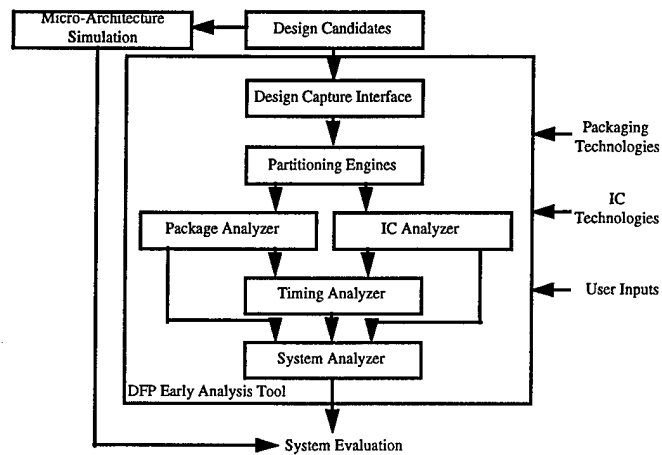
Microelectronic Systems--University of Tennessee

DESIGN SPACE EXPLORATION USING DESIGN-FOR-PACKAGEABILITY EARLY ANALYSIS TOOL

- Exploration of the multi-dimensional design space can be facilitated by applying partitioning at the early stages of product development.
- The prospective design may be input at different levels of abstractions including conceptual, behavioral and structural representations.
- Important features of the packaging technology (bonding and substrate) are included in the partitioning criteria.
- Visualization of the multi-dimensional design space is necessary.

Microelectronic Systems--University of Tennessee

EARLY ANALYSIS TOOL



Microelectronic Systems--University of Tennessee

Optimum System Partitioning Depends Upon Many Performance Factors Driven by the Choice of Packaging Technology

Performance factors

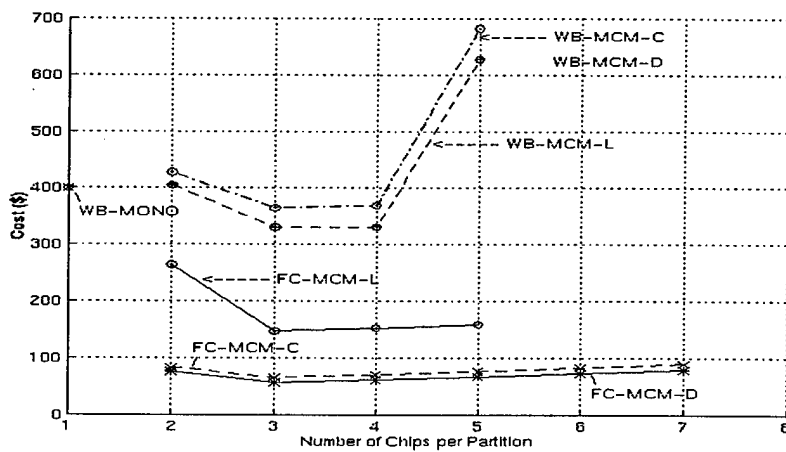
- Cost (Dies, Yield, Bonding, Substrate, Testing, Rework)
- Size (Dies, Interconnect, Substrate)
- Power (Dies, On-Module Drivers, Off-Module Drivers)
- Thermal (External Thermal Resistance Required)
- Noise (Simultaneous Switching, Frequency of Operation)
- Delay (Dies and Interconnect and Package)

Packaging technology choices

- Bonding Type (Peripheral: Wire-Bond, Area: Flip-Chip)
- Substrate Type (MCM-L, MCM-C, MCM-D)

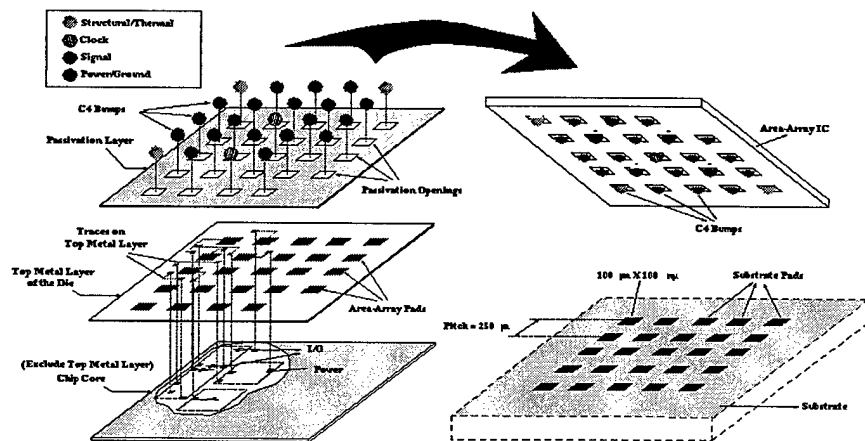
Microelectronic Systems--University of Tennessee

System Cost Analysis



Microelectronic Systems--University of Tennessee

AREA ARRAY PAD ROUTING



Microelectronic Systems--University of Tennessee

ACCOMPLISHMENTS:

Design for Packageability Tool

- **Developed Design for Packageability (DFP) tool to perform design space exploration/early analysis of million-transistor, multi-chip systems including:**
 - **SUN Microsystems MicroSparc CPU**
 - **SUN Microsystems MicroSparc CPU plus cache**
- **Developed Multi-Objective Design Advisor (MODA) to visualize trade-offs in the design space.**
- **Synthesized and performed physical layout of our own version of the MicroSparc CPU for both monolithic and multi-chip cases.**

Microelectronic Systems--University of Tennessee

ACCOMPLISHMENTS:

Area Array Activities

- **Working on automating the physical design of the area-array ICs.**
- **Working with CASCADE to empower the EPOCH tool with the area-array layout capability.**
- **In process of evaluating/helping the single-die bumping services offered by Rockwell & MCNC through the MIDAS.**
- **Coordinating efforts with MMS to evaluate their flip-chip assembly for possible fabrication of the MicroSparc MCM.**

Microelectronic Systems--University of Tennessee

ACCOMPLISHMENTS: Design & Development of a DSP/MCM

- Obtained the MMS & IBM & nCHIP MCM-C/D design-kits.
- Completed the physical layout of a DSP/MCM.
- MCM has been fabricated by MMS.
- MCM is in process of being tested.
- Exercised the MIDAS MCM services.
- Mentor Graphics MCM-Station tutorial has been created and will be placed on line for other universities.
- Assessed the availability MCM technology for Universities.
- Working with other ARPA contractors on a faculty MCM course.

Microelectronic Systems--University of Tennessee

DSP-MCM Project

Motivation :

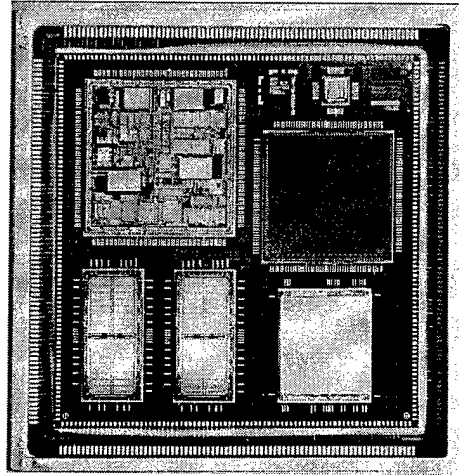
- Experience MCM design techniques , CAD tools and prototyping.
- Enhance the probability of success of the MCM-MicroSparc project.
- Exercise MIDAS MCM service.
- Provide educational example.
- Assess the availability of MCM technology to universities.

Description :

- Motorola 32-bit floating-point DSP
- Multiprocessing capability
- 256K SRAM & 128K FLASH
- 6 channel 12-bit ADC
- Xilinx 4010 FPGA with 10K gates and 160 signal I/Os
- Packaged in a 37mm, Ceramic QFP, 320 pins, 255 I/O

Microelectronic Systems--University of Tennessee

UTK DSP MCM



Microelectronic Systems--University of Tennessee

Where Are We Going ?

- Complete the design & fabrication of the MicroSparc MCM.
- Extend and enhance the Design for Packageability tool .
- Complete automated area-array pad route .
- Empower system designers, especially designers of multi-million transistor ICs, to obtain optimized MCM-based systems.

Microelectronic Systems--University of Tennessee

Dehkordi, P., Ramamurthi, K., Bouldin, D., Davidson, H., and P. Sandborn,
“Impact of MCM Technology on Microelectronic Systems Partitioning: A Case Study”,
Proceedings of 1995 IEEE Multi-Chip Module Conference, Santa Cruz, CA, February 1-2, 1995.

Impact of Packaging Technology on System Partitioning: A Case Study

Peyman Dehkordi, Karthi Ramamurthi, Don Bouldin[†]
Howard Davidson[‡] and Peter Sandborn[§]

Abstract

This paper emphasizes concurrent consideration of the partitioning of a microelectronic circuit design into multiple dies and the selection of the appropriate packaging technology for implementation of the entire system. Partitioning a large design into a multichip package is a non-trivial task. Similarly, selection of the MCM packaging technology to accommodate a multichip solution can also be puzzling. The interdependencies of these two problems afford the opportunity to achieve a global optimum when considered concurrently. In this paper we address the partitioning/MCM technology tradeoff, their interdependency, and previous work in this area. The SUN MicroSparc CPU is used as a demonstration vehicle and is partitioned for different MCM technologies. The preliminary results show that the optimum number of partitions and contents of each partition depend heavily on the choice of MCM technologies for a given application.

1 Introduction

Multi-chip modules have been gaining popularity and becoming more available during the past few years. The designer is now faced with a variety of MCM packaging technologies and has to understand and compare them for a given application. Conventionally, this has been performed by the package designer mainly toward the end of the design cycle. However, to achieve a more nearly optimum system, packaging-related issues **should** be considered throughout the design cycle by system, IC, and

package designers. About 40% of the the product cost is determined by the decisions made in the first 10% of the design cycle [1]. This suggests that the choice of packaging should be explored at the early stages of the design for a more globally optimum system. Considering the critical packaging issues early in the design cycle is termed "Design for Packageability" (DFP) and is discussed in [2].

Comparisons between different MCM technologies cannot be made by just considering the physical and electrical parameters of the technologies. True comparisons should be made at the system level by understanding the impact of the different MCM technologies on the overall cost/system performance of the final system.

2 Problem Definition and Motivation

As a part of our research, we are trying to identify the stages in the design cycle which will benefit the most from taking advantage of the new solutions offered by MCM packaging. Figure 1 shows how partitioning drives all the system performance parameters such as system cost, size, thermal, power, packaging delay and simultaneous switching noise. Chip bonding and substrate technologies determine the physical constraints on the partitioning process. It can be seen that the choice of the packaging technology propagates through the partitioning process and then impacts system performance. Thus, there is a need to explore the effect of the various packaging technologies on system partitioning and hence system performance to achieve an optimum system. The main objective of our work is to study the interaction and impact of the various bonding and substrate technology alternatives on system partitioning and performance. Partitioning an ultra-large single die into multiple smaller dies housed on a MCM is used as an example; however, this approach can be applied to a larger class of applications using multiple levels of packaging hierarchy.

This research was sponsored in part by the Advanced Research Projects Agency under ARPA order A879 and monitored by the Army Research Office under grant DAAH04-94-G-0004 to the University of Tennessee.

[†] University of Tennessee, Knoxville, TN

[‡] Sun Microsystems, Mountain View, CA

[§] MCC, Austin, TX

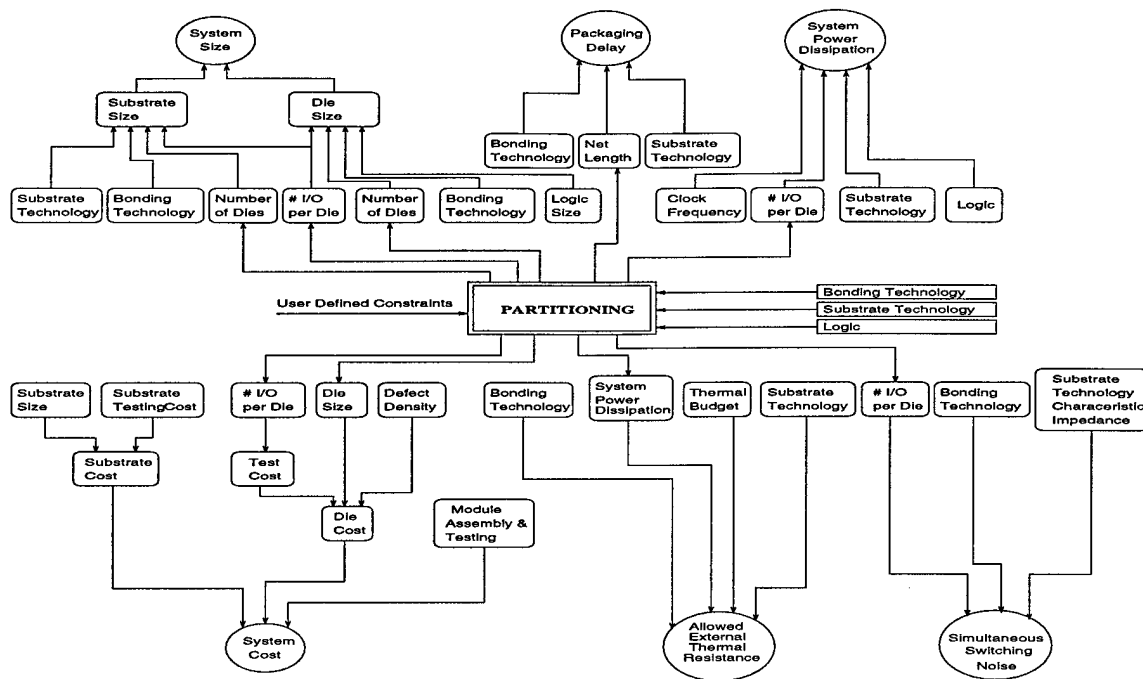


Figure 1: Interaction between System Performance Parameters and Partitioning using DFP.

The impact of bonding technologies such as area array (Flip-Chip, FC) and peripheral (Wire-Bond, WB) on the die size and layout of VLSI dies is discussed in [3]. That work was focused at the physical die layout level rather than the system level. A more conceptual trade-off analysis between peripheral and area array bonding in MCMs is presented in [4]. The design is described in terms of gate counts used Rent's rule to establish the number of I/Os for a given circuit size. Our work tries to establish the interaction of the various packaging alternatives, partitioning and the system performance when more information is available about the design. The SUN MicroSparc is used as a demonstration vehicle to illustrate the extent of the interaction. The design is described at the functional unit level consisting of the following: integer unit (IU), floating point unit (FU), memory management unit (MMU), data cache (D-CACHE), instruction cache (I-CACHE), S-bus controller (S-BUS-CTL), memory interface unit (MEM-INTF), clock control and buffers (CLK-CTL), and miscellaneous control logic (MISC) as shown in Figure 2.

3 Approach

The intent of this work is not to promote a new partitioning scheme since numerous techniques

have already been described [5],[6]. Our approach is to develop a framework in which various packaging/partitioning choices can be explored and evaluated concurrently. Since performing a detailed evaluation at the system level for various packaging/partitions can turn into a time-consuming task, we have employed an estimation-based, early-analysis technique. At this level there are only nine functional units so the number of possible candidates is low enough (approx. 21,000) that it is possible to search the solution space exhaustively. This guarantees the best solution will be found. At the register transfer level where there are a large number of components, it is more appropriate to use partitioning techniques based on algorithms such as simulated annealing which can guarantee a good solution among several possible solutions.

The steps involved in our concurrent analysis are shown in Figure 3. The user specifies the package and die related information (i.e. bonding technology, maximum die size). The constraint generator derives the actual constraints from these user specifications. The exhaustive partitioner then generates all possible candidate partitions. The algorithm used to generate partitions is described in [7]. The die size and die I/O estimates are then calculated for each of the partitions. Next the partitions are verified against the constraints. These constraints are used to qualify a partition for further processing. Further

processing involves estimating the system performance characteristics such as system cost, system size, module power, allowed external thermal resistance and total simultaneous switching noise in the module. The MSDA (Multichip System Design Advisor) tool developed by MCC is used to estimate the system performance characteristics.

4 Results

We have concurrently considered the following:

- a) Wire-Bond/MCM-C
- b) Wire-Bond/MCM-D
- c) Wire-Bond/MCM-L
- d) Flip-Chip/MCM-C
- e) Flip-Chip/MCM-D
- f) Flip-Chip/MCM-L

The exhaustive partitioner has generated over 21,000 partitions for each type of packaging but only those partitions that meet the die and package constraints have been considered for analysis. The die parameters provided by the user are given in Table 1.

Property	Value
Signal/Ground (peripheral)	4
Signal/Ground (area array)	6
Bond pad size (peripheral)	200 * 200 (microns)
Bond pad size (area array)	125 * 125 (microns)
Min Bond Pad pitch (peripheral)	200 (microns)
Min Bond Pad pitch (area array)	250(microns)
Wafer Diameter	6 inches
Unusable Wafer Border	0.4 inches
Wafer defect density	3 defects per sq.inch
Processed Wafer cost	\$800
Wafer Bumping cost	\$200
Defects due to Wafer Bumping	0.2 defects per inch
Die Test cost	\$0.01 per I/O

Table 1: Die Assumptions Provided by the User.

The result of the cost analysis is shown in Figure 4(a) which displays those partitions satisfying the given constraints with the lowest system cost for a particular number of dies in the die set. The

FC/MCM-D design offers the lowest overall system cost for implementing this particular application in a MCM. The system cost is comprised of the die, bonding, substrate and assembly cost. The substrate and assembly cost estimates used in this analysis is discussed in [4]. The flip chip design offers higher I/O count and takes full advantage of the higher interconnect density of the MCM-D. The combination of these two choices reduces the die area (and hence the die cost) considerably as compared to the conventional peripheral wire-bond design. It should be noted that FC/MCM-D is not highly sensitive to the number of chips in the partition.

The multichip design implemented in WB/MCM-C and WB/MCM-D exhibit the highest overall system cost. The lower I/O count offered by the peripheral wire-bond design results in larger die area which results in reduced yield and higher die cost.

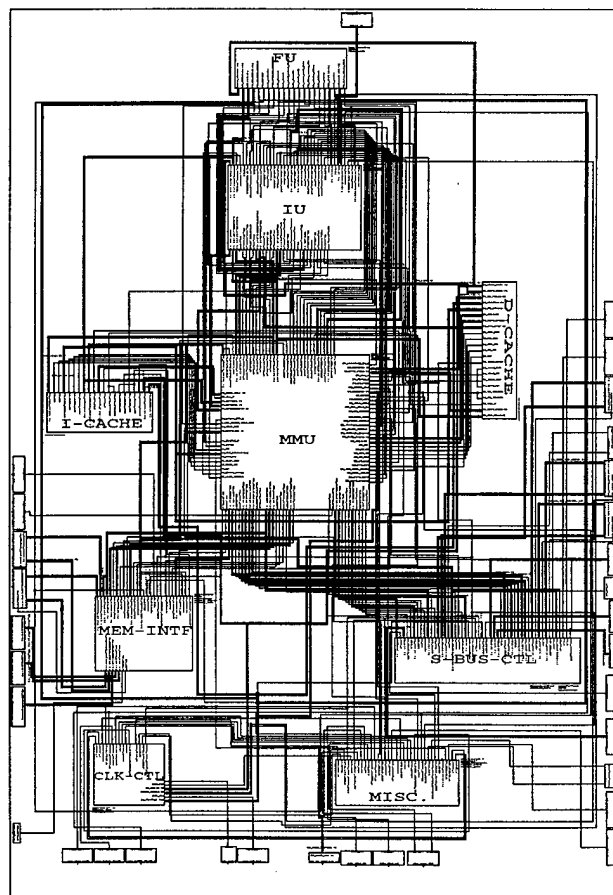


Figure 2: Functional-Level Diagram of the MicroSparc.

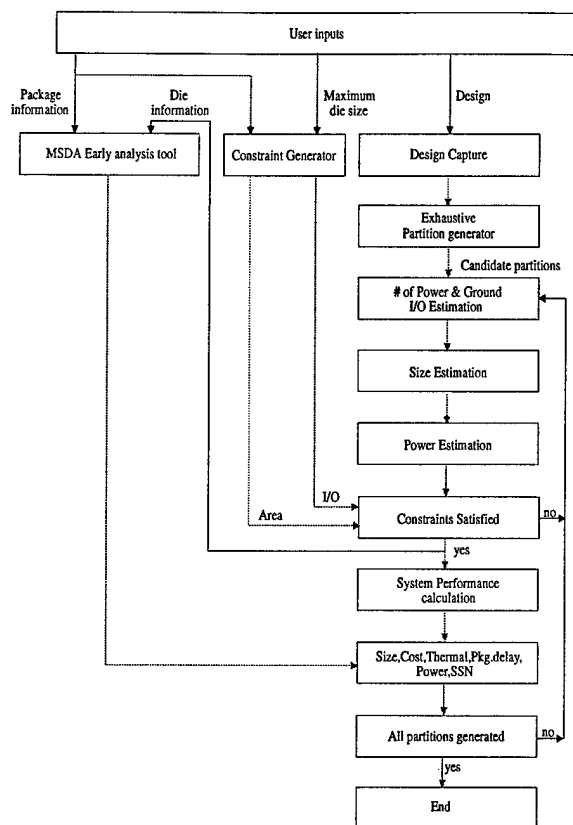


Figure 3: Block Diagram of Exhaustive Partitioning.

For this application, the die cost in the wire-bond case dominates the substrate and assembly costs. Therefore, the die set which offers the lowest system cost is the same for wire-bond design using any of the three substrates. However, this is not true for the flip-chip designs since their die costs are comparable to the substrate and assembly costs.

Figure 4(b) shows the size of the partitions which offer the lowest system cost. The flip-chip design using MCM-D exhibits the smallest module size. This is due to the combination of the reduction in die area because of area-array bonding and the reduction in substrate size with the use of MCM-D interconnect.

A measure of the simultaneous switching noise analysis is shown in Figure 4(c). The noise data shown corresponds to the partitions having the lowest system cost. The flip-chip designs have lower inductance and hence provide lower switching noise.

Figure 4(d) shows the total power dissipation of the modules which have the lowest system cost. The flip-chip/MCM-D designs have higher power dissipation compared to the wire-bond designs since the flip-chip designs have more I/Os. The flip-chip/MCM-C has the worst power dissipation due to the higher interconnect capacitance of the substrate. In this particular application, the power dissipation increases with the increase in number of chips due to the increase in the number of outputs in the die-set.

The results from the thermal analysis is shown in Figure 4(e). The worst-case external thermal resistance of the die in the partition is heavily dependent upon the total power dissipation in the module. Higher values of external thermal resistance indicate less power dissipation inside the MCM. Thus, the external thermal resistance decreases with the increase in the number of chips in the partition. There are some versions of flip-chip/MCM-D where special process techniques (e.g. potting and lapping the completed assembly) result in better external thermal resistance characteristics.

Figure 4(f) shows a figure-of-merit for packaging delay of these MCM systems. The delay was computed for a length equal to the diagonal length of the module. The interconnect line was modeled as either lumped RLC or a transmission-line based on their lengths. Each line was terminated and a total of eight receivers were assumed for each driver. The delay calculations include time-of-flight, RC charging and reflections and, therefore, are a function of the dielectric constant and size of the MCM module. For the monolithic case, the delay was calculated for an interconnect signal line within the die with a length equal to the diagonal length of the die.

5 Summary and Conclusions

Each type of MCM technology has a different cost/performance characteristic. It is important to evaluate these technologies for the specific application in hand for the best price/performance. Evaluation and selection of these technologies should not be solely based on the physical and electrical characteristics of the technology itself but should be based on price/performance of the entire system by considering the interdependency of MCM technologies and partitioning at the system level.

The performance parameters of cost, size, power, thermal, simultaneous switching noise and package delay for the six different packaging alternatives are shown in Table 2. The candidate ranking was arrived by considering an overall figure of merit of the various

	Monolithic	WB MCM-L	WB MCM-C	WB MCM-D	FC MCM-L	FC MCM-C	FC MCM-D
System Cost (\$)	400.05	330.70	365.17	364.94	147.46	66.18	57.45
System Size in^2	0.3488	1.34	1.34	1.34	0.9	0.91	0.6
Module Power (W)	4.9	5.0579	5.1162	5.0388	5.1946	5.7227	5.6963
Ext. Therm. Res. (degC/W)	12.69	11.45	11.05	11.59	10.52	8.06	9.63
SSN	124	410	494.17	476.03	6.85	8.07	7.77
Pkg. Delay (ns)	0.7918	1.3229	2.1792	1.3806	1.2134	1.9289	1.1459
Ranking		4	6	5	2	3	1

Table 2: Comparison of System Parameters for Bonding and Substrate Technologies.

Chip	Pins	Area (mm ²)	Modules
1	485	49.428059	D-CACHE, I-CACHE, MMU
2	298	45.510590	FU, IU
3	414	28.451555	MEM-INTF, SBC, CLK-CTL, MISC.

Table 3: Contents of the Best Overall Partition.

system performance parameters. The best partition, consisting of three dies, is shown in Table 3.

For this particular application, the results indicate that the overall system cost would be reduced by a factor of seven if the single-chip CPU were divided into three chips, bonded using flip-chip technology and interconnected on an MCM-D substrate.

To date, the functionality of the partitions has not been considered by the partitioning tool. There is still a need for an experienced system architect designer to compare the results for the best design architecture. We plan to analyze the cost/performance of the different cache sizes added to the design and perform the detailed analysis of the above candidate designs to verify the validity of the model used in the analysis.

The methodology of partitioning with DFP in mind is applied here to a design described in the functional unit level. We plan to extend this concept to designs described at the behavioral and structural (RTL) levels as well.

References

- [1] Kuk, D., "Examining the Impact of DFM on Product Development," *Electronic Packaging and Production*, Vol. 33, No. 5., pp. 36-4 to 36-7, May 1993.
- [2] Dehkordi, P. and D. Bouldin, "Design for Packageability: Early Consideration of Packaging from a VLSI Designer's Viewpoint," *Computer*, vol. 1, pp. 76-81, April 1993.
- [3] Dehkordi, P. and D. Bouldin, "Design for Packageability: The Impact of Bonding Technology on the Size and Layout of VLSI Dies," *Proc. IEEE Multichip Module Conf.*, pp 153-159, March 1993.
- [4] Sandborn, P., Abadir, M. and C. Murphy, "The Tradeoff Between Peripheral and Area Array Bonding of Components in Multichip Modules," *IEEE Trans. on Components, Packaging and Mfg. Tech. - Part A*, vol. 17, no.2, pp. 249-256, June 1994.
- [5] Shih, M., Kuh, E. and R. Tsay, "Performance Driven System Partitioning on Multichip Modules," *Proc. 29th Design Automation Conference*, pp 53-56, June 1992.
- [6] Vemuri, Ram., Kumar, N. and Ranga Vemuri, "Two Randomized Algorithms for Multichip Partitioning Under Multiple Constraints," Tech. Report TM-ECE-DDE-94-36, Univ. of Cincinnati, 1994.
- [7] Even, S. *Algorithmic Combinatorics*, The Macmillan Company, 1973.

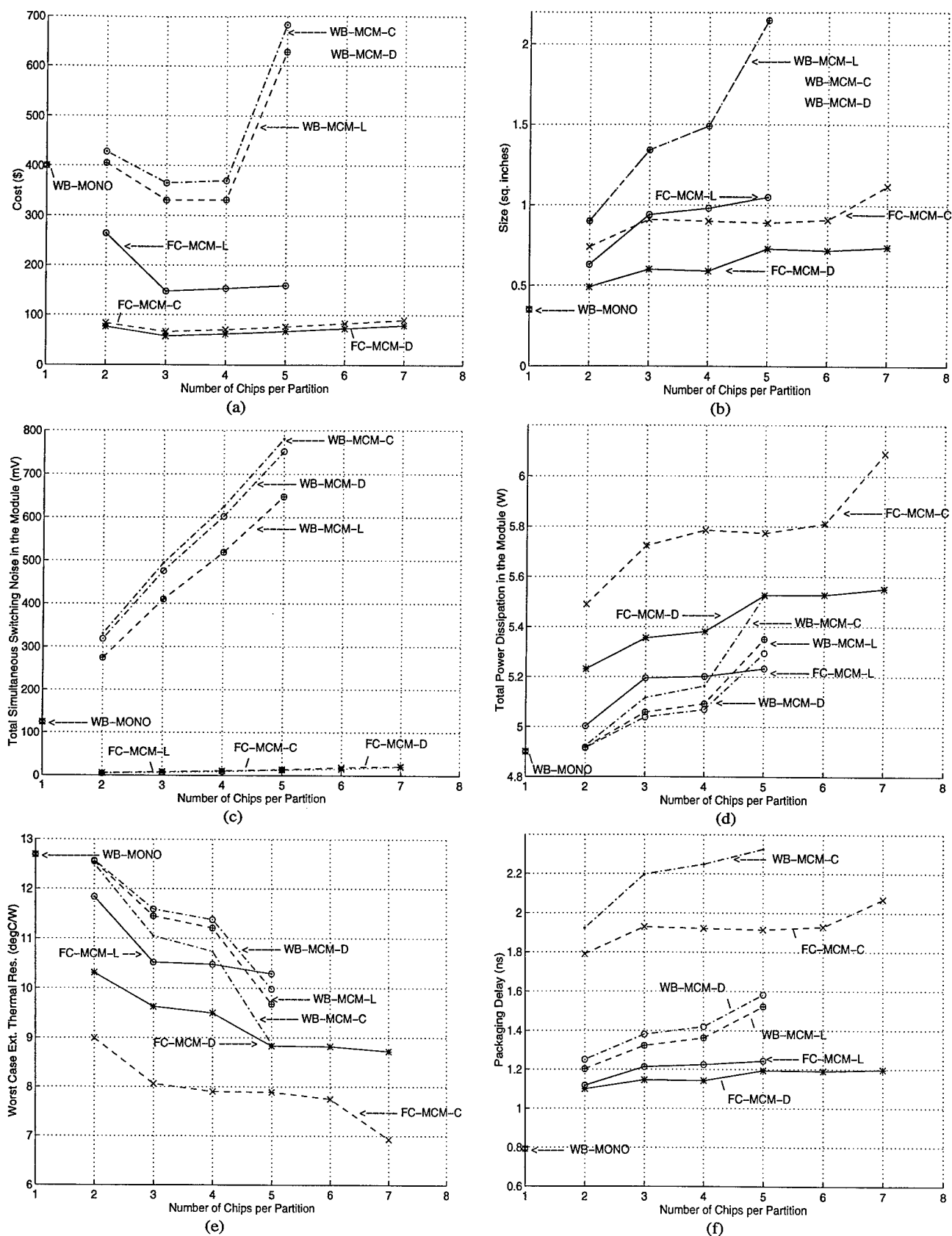


Figure 4: Results of the System Performance Analysis.

Bouldin, D. and P. Dehkordi,
“Design for Packageability: An Overview”,
Proceedings of 1995 International Intersociety Electronic Packaging Conference, Lahaina, HI,
March 26-30, 1995.

DESIGN FOR PACKAGEABILITY: AN OVERVIEW

Donald W. Bouldin
Electrical & Computer Engineering
University of Tennessee
Knoxville, Tennessee

Peyman H. Dehkordi
Electrical & Computer Engineering
University of Tennessee
Knoxville, Tennessee

ABSTRACT

To achieve higher performing systems, microelectronic system designers should consider the effects of packaging while developing integrated circuits (ICs). This practice, which we call "Design for Packageability", should permit a global optimization of the entire system by incorporating critical packaging factors into the IC design flow. Managers and engineers must be able to perform an early analysis of the expected system performance and cost prior to committing major engineering resources and capital to its development. Early analysis computer-aided design (CAD) software is being developed to show the effects of various IC design alternatives which may have even more impact on overall system performance than the change from conventional packaging to multi-chip module (MCM)-based packaging.

INTRODUCTION

For several years, the development of higher performing supercomputers spurred IBM to develop MCM technology which uses ICs with area pads. The bonding procedure used by IBM is known as C4 (controlled-collapse chip connection) in which pads may be located within the area consumed by the IC and not just on its periphery. Not only do area pads provide the IC designer with more I/O, and hence increased bandwidth when communicating with other ICs on the module, but it also increases the testability of the IC while reducing its size, power consumption and cost. Now that IBM and others are making this MCM-based technology available to non-IBM designers, there is a need for IC designers to alter their mind-set to design ICs which use area pads. Thus, new trade-offs will be presented to the IC designers and these changes in packaging technology should be incorporated into the CAD tools that the IC designers use.

This specific example falls within a larger class of design which we have chosen to call "Design for Packageability".

Just as the practice of Design for Testability resulted in overall better systems, we believe a similar impact can be made by following this procedure in the case of MCM-based systems. Until Design for Testability was followed, many IC designers made trade-offs which resulted in almost impossible problems for test engineers to solve. By incorporating a few features early in the design process, the resulting ICs were more easily and readily tested. This same result should be experienced with Design for Packageability in that a more globally optimum system should be developed if IC designers think ahead and make trade-offs with critical packaging parameters in mind.

Moreover, this concept can be extended to the general problem of performing optimization on separate domains such as design and technology. If each domain is treated in isolation, then suboptimal systems will result. Whereas treating a proper mix of the domains simultaneously can lead to globally optimum systems.

In this paper, we first present some preliminary results of applying the Design for Packageability methodology. The specific case considered is the impact of bonding technology on the overall system performance. Next, the goals of our research project will be described as a means of proving and advancing this methodology. Finally, conclusions are presented to indicate the importance of this work.

THE NEW METHODOLOGY

The design flow for this new methodology can be visualized as shown in Figure 1. In the past, IC designers have designed their chips thinking that each one would be housed in a single package and then interconnected on a printed circuit board (PCB). With the availability of MCM technology, system designers are initially taking the same chips (or not changing the traditional design flow) and packaging them in an MCM. The expected increase in system performance is generally achieved since the large capacitances involved in driving traces on a printed circuit board have been removed

and the number and density of the interconnections have improved dramatically.

However, following this traditional design flow does not fully exploit the capabilities of the MCM technology. Only when the IC designer realizes the potential advantage of designing his chips for the bonding that is available in the new technology is the system fully optimized.

For comparison purposes, consider three systems as illustrated in Figure 2. System 1 consists of peripheral-pad ICs which are wire-bonded onto a conventional PCB substrate. This serves as an example of today's typical systems. A normalized performance factor of 1.0 (i.e. a measure of power, size, speed, etc.) is assumed for this system for comparison purposes. System 2 (following the traditional design flow) utilizes the same set of ICs. Hence, these ICs are wire-bonded onto a MCM substrate. This is representative of the second class of systems that is being widely used as a quick boost in system performance by moving from a PCB to a MCM. The gain in the performance is completely due to the change in substrate. An assumed value of 20 % is shown in Figure 2.

Following the new design for packageability methodology, the third system consists of a new set of area-pad ICs which are C4-bonded onto a MCM substrate. Calculations indicate that System 3 will outperform the other two systems because of the change in the integrated circuits. A representative value of 30 % is shown in Figure 2.

Published reports indicate that this new methodology is not currently being practiced. Reports in the literature can be divided into two areas: experimental and global. The experimental results show the impact of packaging by fabricating a system based on two or more packaging techniques [1, 2]. They compare conventional packaging with MCM-based packaging. The ICs (dies) remain the same in either package and all comparisons are done at the system level. No formal discussions are given and the change in the system performance is due completely to the change in the packaging environments.

The global results reported to date include SUSPENS, AUDiT, and MSDA-PKG [3-5], which use formal discussions to describe the impact of packaging. They approach the issues by using a mathematical model to describe the system in terms of the package and the ICs. This model is then used to describe the impact of the packaging in the overall system performance. The packages are described by their physical and electrical characteristics. The ICs, however, are modeled globally by the physical sizes of the dies and the number of I/Os.

All of the above tools consider only the effect on system performance provided by the package. None of these methods takes into account the impact of the die on system performance. They all assume that the performance of the die is independent of the choice of packaging.

Recent work by the Microelectronic Systems Group at the University of Tennessee has investigated the effect on system performance of the die for two different packaging technologies [6]. As a proof of principle, an image processing chip-set was designed based on the Massively Parallel Processor (MPP). This chip-set was designed, simulated, and synthesized for conventional peripheral pads (wire-bondable) and for area pads. The preliminary results demonstrated how the performance of the dies change as a result of changes in the design being targeted for peripheral pad bonding and area pad bonding. The impacts of the packaging technology

on the performance of the dies have been demonstrated quantitatively in the following areas: power, size, testability, and cost. In the case for size, the effect of the package on system performance was 21 % and the effect of the die was 20 %. Evidently, changing only the package did not result in an optimum system. It took a change in the design of the IC to achieve the full benefits of the new MCM-based technology.

SYSTEM OPTIMIZATION NEEDS

In light of the potential this methodology has for enhancing the overall system performance of MCM-based systems, a variety of needs become apparent. These include:

1. MCM-based system optimization should be investigated, demonstrated and documented primarily by incorporating critical packaging factors into the design of the integrated circuits.
2. The procedure for using area pads rather than peripheral pads should be utilized, documented and promulgated publicly as part of a Design for Packageability Procedures Manual.
3. A comprehensive system model that can be used to analyze and compare from a system designer's viewpoint the impact of different packaging technologies and different IC design methodologies should be developed.
4. The effect on system performance and cost of changing from PCB-based to MCM-based packaging should be demonstrated.
5. Similarly, the effect on system performance and cost of changing from ICs using peripheral pads to those using area pads should be demonstrated using two MCM-based systems.
6. Early analysis CAD software incorporating such a system model should be developed and made available to others.
7. Return on investment calculations should be made to assist managers and engineers in deciding when it is appropriate to use area pads rather than peripheral pads.
8. Commercial CAD software for designing ICs should be enhanced to include consideration of critical packaging parameters.
9. Seminars and short courses should be organized to disseminate the results of these studies and demonstrations.

Portions of the above tasks are presently being undertaken by a team of researchers at the University of Tennessee, SUN Microsystems and MCC. Conventional approaches to microelectronic system design, partitioning and physical IC design will be optimized to take full advantage of MCM packaging. The purpose of this project, which is sponsored by the Advanced Research Projects Agency, is to assist the system designer in exploring the design space in order to perform this optimization. This goal will be accomplished by identifying the areas of the design cycle which benefit the most from exploiting the new opportunities which MCMs provide.

In support of these objectives, the following will be developed:

1. Early Analysis Tool: This tool will allow the system designer to explore IC and package options at an early stage of the design which may be input at different levels of abstractions including conceptual, behavioral and structural representations.
2. Simultaneous IC and MCM Physical Design Tool: This interactive tool will enable the system designer to consider the placement of the ICs on the MCM substrate before completing the IC pin assignment and physical design.
3. Optimized Partitioning Tool: This tool will partition a given design at different levels of abstractions. It will differ from other partitioning tools in that it will consider salient features of the chosen MCM technology.
4. Multi-objective Design Advisor: This information management tool will empower the designer with an efficient way of analyzing results and performing trade-off studies.
5. Application of Design for Packageability: The tools described above will be utilized to analyze and partition an industrial-strength system (SUN MicroSparc CPU) into multiple ICs that will be fabricated and tested as a single MCM.

CONCLUSIONS

Multi-chip integration technology has the potential to increase significantly overall system performance and reliability while reducing size, weight, power and cost. The work described here explores and demonstrates the impact of these MCM-based technologies on IC design and, hence, on the overall system. Development and verification of the planned Design for Packageability procedures should accelerate the architectural exploitation of both ICs and MCMs and result in higher performing systems.

Armed with this knowledge, system designers could answer the following types of questions:

1. What are the costs and benefits of using area pads instead of peripheral pads when designing ICs to be packaged in a MCM?
2. What architectural changes might be made that would result in significant enhancements in system performance and cost?
3. Is it worthwhile to redesign existing peripheral pad ICs to be area pad ICs?
4. What changes need to be made to commercial CAD software to include these critical packaging factors?

REFERENCES

- [1] Shiao, J. and D. Nguyen, "Performance Modeling of a Cache System with Three Technologies: Cyanate Ester PCB, Chip-on-Board, and CU/PI MCM", *Proceedings of 1992 IEEE Multi-Chip Module Conference*, Santa Cruz, California, pp.134-137, March 1992.
- [2] Bartlett, C., Segelkton, J. and N. Teneketges, "Multi-Chip Packaging Design for VLSI-Based Systems", *IEEE Transactions on Components, Hybrid, and Manufacturing Technology*, Vol. CHMT-12, No. 4, pp. 647-653, December 1987.
- [3] Bakoglu, H. B., "Circuits, Interconnections, and Packaging for VLSI", Reading, MA: Addison-Wesley, 1990.
- [4] Krusius, J., "Packaging Architecture Considerations of High Density Multi-Chip Electronic Packages via System Optimization", *Transactions of ASME, Journal of Electronic Packaging*, vol. 112, p. 267, 1990.
- [5] Sandborn, P., Ghosh, R. and K. Drake, "Computer-Aided Conceptual Design of Multi-Chip Systems", *Proceedings of the 1993 Custom Integrated Circuits Conference*, pp. 29.4.1 - 29.4.4, May 1993.
- [6] Dehkordi, P., "Design for Packagability: The Impact of Bonding and Interconnect Technologies on the Performance of VLSI Dies", Ph.D. Dissertation, University of Tennessee, 1992.

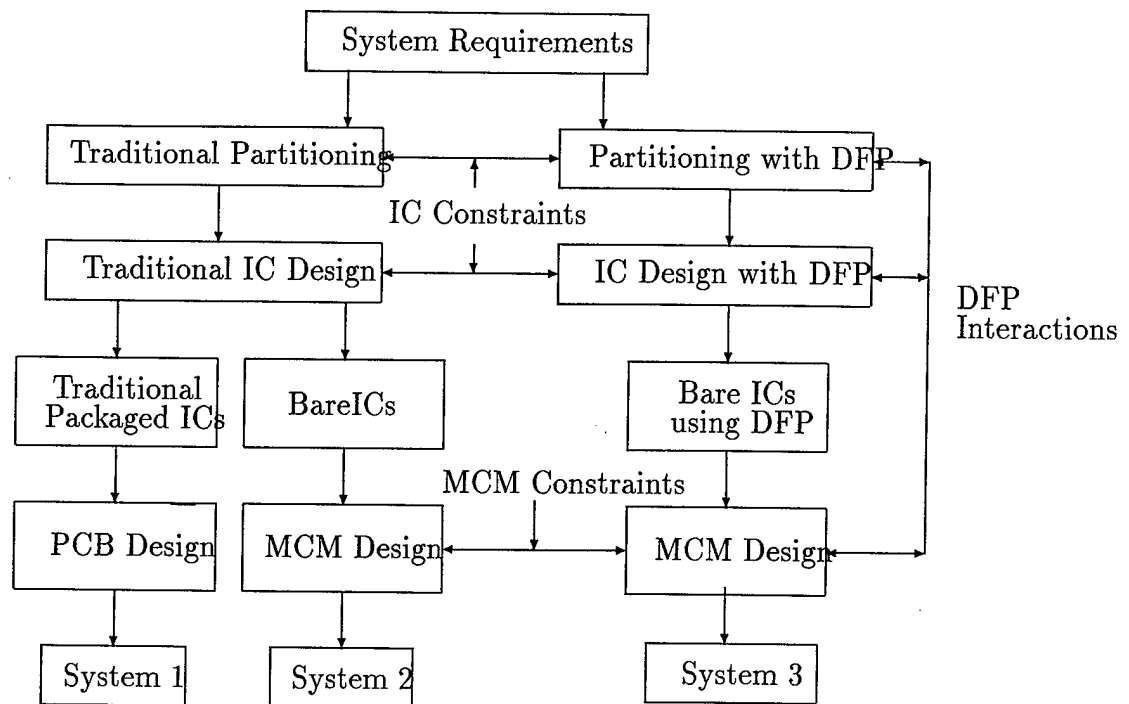


Figure 1: *Design for Packageability Methodology.*

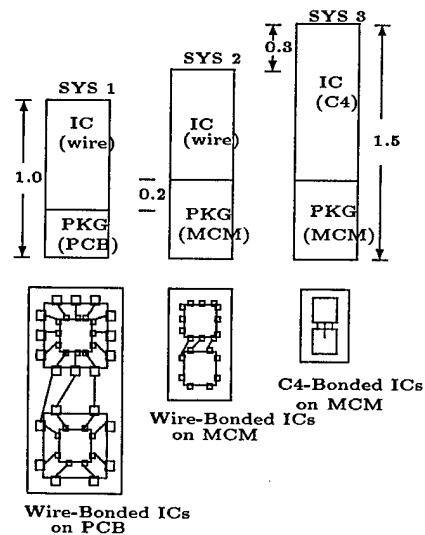


Figure 2: *Change in System Performance due to Package and IC changes.*

Chattapadhyay, S., Dehkordi, P., and D. Bouldin,
“An Overview of Placement and Routing Algorithms for Multi-Chip Modules”,
International Journal of High Speed Electronics - Special issue on CAD for Multichip modules,
December, 1995.

An Overview of Placement and Routing Algorithms for Multi-Chip Modules

Subhomoy Chattopadhyay, Donald W. Bouldin and Peyman H. Dehkordi

Electrical & Computer Engineering

University of Tennessee

Knoxville, TN 37996-2100

Correspondence Author : Prof. Donald W. Bouldin

TELEPHONE : 615-974-5444

FAX : 615-974-5492

Email : bouldin@microsys1.engr.utk.edu

Abstract

Multi-chip modules are now required to achieve higher system speed and greater density than the traditional single chip packages mounted on printed circuit boards. Algorithms for placement of bare dies and routing of their interconnections on MCM substrates are reviewed in this paper. Comparisons are given to point out the strengths and weaknesses of each approach. This information can assist researchers in identifying those areas which need improvement and application designers in selecting the most appropriate algorithm for a specific application.

Keywords : Placement, routing, multichip, modules, algorithms, die

Contents

1	Introduction	4
2	Physical Design of MCMs	5
3	MCM Placement Algorithms	7
3.1	Performance-Driven Placement for MCMs	7
3.2	Conventional MCM Placement Approaches	8
3.3	Placement Using Multiple Design Criteria	9
3.4	Comparison of Different Placement Approaches	10
4	MCM Routing Algorithms	10
4.1	Introduction to the MCM routing problem	10
4.2	Maze Routing	11
4.3	Multiple-Stage Routing	12
4.4	Integrated Pin Redistribution and Routing	13
4.5	Gridless MCM Router based on Rubber-band Sketches	15
4.6	Performance-Driven Global Routing	16
4.7	Crosstalk Minimization Routing	21
4.8	Three-Dimensional Routing	22
4.9	Comparison of the Different Routing Algorithms	23
5	Conclusions and Scope for Future Work	23
6	Acknowledgments	26

List of Figures

1	<i>MCM Physical Design - Flow</i>	35
2	<i>Some Delay Models</i>	36
3	<i>A - Tree vs Steiner Tree</i>	37
4	<i>Zero - Skew Tree Construction</i>	38
5	<i>A tapered interconnect</i>	39
6	<i>A Multi Stage Clock Tree Distribution</i>	40

List of Tables

1	Comparison of Different MCM Placement Algorithms	41
2	Comparison of MCM Routing Algorithms, Part I	42
3	Comparison of MCM Routing Algorithms, Part II	43

1 Introduction

MCM physical design consists of two operations: placement and routing. *Placement* refers to the positioning of unpackaged or bare dies on the MCM substrate such that one or more figures of merit are optimized. Typical criteria include netlength minimization, proper heat distribution and overall system timing performance. *Routing* is the task of interconnecting the dies on the substrate such that one or more figures of merit are optimized. For routing, typical optimization criteria include netlength, via, layer or crosstalk minimization.

These two operations are mutually dependent, but the size and complexity of practical problems generally dictate that each one be solved separately. Each operation is order-dependent in that the placing of one die or the routing of an interconnection leave fewer options for the remaining dies to be placed or interconnections to be routed. Thus, those placements or interconnections which are considered to have a greater impact on the optimization criteria are performed first.

MCM physical design requires consideration of many issues which cannot be handled satisfactorily with computer-aided design tools developed for integrated circuits or printed circuit boards [49, 42, 12, 16]. Factors like the number of layers in the MCM substrate, the type of bonding and the type of substrate (ceramic, laminate or deposited) must be an integral part of the optimization procedure [39]. Considering the packaging issues early in the design cycle leads to better physical design and overall superior system performance [12].

An overview of the MCM physical design process is presented in the next section. In Section 3, MCM placement algorithms are detailed along with a comparison of their strengths and weaknesses. In Section 4, MCM routing approaches are presented and compared in a similar manner. The final section presents some general conclusions and outlines some areas for future

research.

2 Physical Design of MCMs

Figure 1 illustrates the operations involved in MCM physical design with respect to the overall design flow. The top-level hardware description language (VHDL or Verilog) is synthesized into a structural netlist of library components and their interconnections. For a very large design the netlist can then be partitioned into a set of different dies based on certain constraints such as area, thermal, delay and testability. This step is known as MCM partitioning or system partitioning. The goal of partitioning is to improve the functionality and routability of the design. After the layouts for the individual dies have been obtained, the dies have to be placed and then routed on the MCM substrate.

The actual physical design process does not include system partitioning and begins with the placement process. The system partitioning step is eliminated in cases where off-the-shelf bare dies are used as is the case with most MCM designs of today. The first main step in the physical design process is *chip placement*. MCM placement involves placement of dies on the MCM substrate such that the objectives of proper heat distribution, minimization of interchip delays, noise and tolerant load distance are taken into consideration. However when the fabrication issues are considered, additional constraints in the form of net separation and via constraint become important. This is because the fabrication of densely routed designs may result in low fabrication yield [16] or a non-manufacturable design.

The next step in the physical design process is *pin redistribution* [19] which uses a chip layer to improve the substrate routability of the design. The pin redistribution problem can be stated as: after the placement of the dies on the MCM substrate, redistribute the pins using chip layers

such that the total number of pin redistribution layers, crosstalk and the maximum signal delay are minimized. Another objective of pin redistribution is to maximize the number of nets that can be routed in a planar fashion. During pin redistribution each pin is to be assigned to the nearest point on a grid in the pin redistribution layers. Pins are redistributed uniformly with sufficient spacing so that the connections between the nets and the pins can be done without any design rule violations.

The final step in the physical design process is *routing* which usually consists of *global routing*, *layer assignment* and *detailed routing* [42]. The global routing step assigns nets or wires to different routing regions. Whereas the router in the detailed routing step actually finds a geometric path for each net and routes the terminals connected by the net. The primary goal of the global routing step is to reduce the length of the global routing trees since this reduces the area and delay. On the other hand, the layer assignment problem, which is in itself an NP complete problem, is a step in the routing process where each net is assigned to an x-y layer pair subject to the feasibility of routing the nets on a grid. The layer assignment step is important as it determines the number of layers in an MCM and hence the cost and the cooling mechanism required.

The MCM routing problem is different from the VLSI routing problem primarily due to the larger number of layers available in an MCM (e.g. 60 versus 4). Also the total number of nets and the density of interconnections in an MCM are much larger necessitating the development of a different set of CAD tools for the MCM design process. However at the present time most of the CAD tools for MCM physical design are either repackaged PCB or IC design tools. MCM routing has been considered as a three-dimensional routing process by [41] because the routing is not only done in the two-dimensional plane but also in the vertical direction in the multiple

substrate layers. The number of nets in the design, the substrate size, the number of chips mounted on the MCM substrate and the number of layers will continue to increase progressively necessitating that the MCM routing problem be efficient. The main objectives of MCM routing are reduction of the number of routing layers and the reduction or elimination of crosstalk which finally leads to an improvement in the overall system performance.

3 MCM Placement Algorithms

The MCM die placement problem can be defined as below :

Given a set of chips (dies) C and a set of chip sites S , find a mapping $\phi : C \rightarrow S$ subject to the timing (delay), thermal and area constraints in such a way so as to minimize the number of layers needed for routing.

Hence some of the important goals for the MCM placement tool are: minimize the total area of the substrate, ensure proper heat distribution, minimize the total length of the wire needed for routing and ensure routability of the design in minimum number of routing layers. MCM placement is a difficult combinatorial problem as a good placement involves a tradeoff between these mutually conflicting constraints. Some of the different placement algorithms and approaches are described next.

3.1 Performance-Driven Placement for MCMs

In MCMs some of the netlengths for connection between bare dies can be so long that they have a resistance which is comparable to the resistance of the driver. Such resistances cannot be neglected during placement when the delay due to the nets is estimated. Few approaches

have been developed to perform a delay or performance-oriented placement. Performance-driven placement is very important since the interconnect delays form a major part of the system cycle time. Several performance-driven algorithms based on delay models have been developed [44, 1]. The timing constraints fix the upper bounds on the netlengths and these upper bounds guide the placement process. The delay in multi-terminal nets depends on the net topology as well as the length of the net. The net topology is difficult to estimate at the placement stage of the design process making accurate delay estimation difficult. One way to get over the problem is to consider global routing and placement simultaneously as proposed in [43, 5]. A resistance-driven placement algorithm for placement in multichip modules has been developed by [42]. In this approach the net delay (cost) is modeled as the combination of the delay contributed by the wire that forms the net and the delay contributed by the sink capacitances. The second contribution is calculated based on all driver-to-sink capacitances.

Some researchers have considered the system partitioning and placement as one composite problem [42, 6]. Partitioning-based placement methods tend to spread the wiring across the layout surface and thus produce very routable placements. However it is difficult to model a truly integrated partitioning and placement approach that deals with all the issues involved in system partitioning and also routing. This approach applies for those cases in which the design of the dies has not been finalized yet MCM routing considerations are incorporated.

3.2 Conventional MCM Placement Approaches

The traditional placement methods can be categorized into two groups: *constructive and iterative*. Constructive placement methods take a partial placement as the input and produce a complete placement as the output. On the other hand, iterative placement methods begin

with one initial guess for the placement and then refine this placement taking into consideration certain constraints to obtain a better placement. A detailed description of these techniques is found in [51]. In the iterative placement approach probabilistic search optimization algorithms like the simulated annealing or genetic algorithms are used and the process is iterated until some stopping criteria is satisfied. In each iteration the components are moved around or rotated, and if the new configuration is better than the previous one, then the new configuration is selected. A cost function is invoked to determine the relative merit of each configuration. Some of these iterative methods are described in [51, 13, 17, 7] Esbensen and Mazumder [15] have combined the genetic algorithm and simulated annealing algorithm to speed up the optimization search and obtain better placements compared to either algorithm alone. This approach has been tested for macrocell placement and can also be extended to placement of individual dies on the MCM substrate.

3.3 Placement Using Multiple Design Criteria

Placement of multiple dies on an MCM substrate is a non-trivial task in which multiple criteria need to be considered simultaneously to obtain a true multi-objective optimization. Most researchers have considered only one criteria or objective in the MCM placement algorithms. In most cases it is netlength minimization which results in overall area and delay minimization. An ideal placement tool should result in the smallest layout while conforming to the electrical and thermal requirements. By doing a placement considering both the thermal and netlength criteria simultaneously, it can be seen that the final netlength after optimization is slightly higher. However the heat spread on the MCM substrate is more uniform compared to the case when only netlength minimization is done alone.

3.4 Comparison of Different Placement Approaches

MCM placement is much more complicated than the conventional IC placement problem in VLSI. Even though the number of dies in a MCM is generally well below 100, many interrelated factors determine the final layout quality. Table 1 compares different MCM placement approaches. The netlength minimization approach as followed by different researchers is not sufficient. Accurate delay estimation to account for the resistance of the nets is necessary.

4 MCM Routing Algorithms

4.1 Introduction to the MCM routing problem

The MCM routing problem can be defined as follows:

Given a set of placed chips (dies) and a netlist interconnecting different pins on the chips, route all the nets in such a way such as to use the minimum number of routing layers and satisfy some constraints.

Some of the objectives that need to be considered for optimization are netlength minimization, crosstalk minimization, via minimization and meeting the manufacturability constraints that guarantee that the yield is high and the designs are routable.

As MCM technology develops the number of chips on the MCM substrate and the number of layers will increase dramatically causing the need for very efficient routing algorithms. For example an MCM with 100 chips and 63 layers has been reported in [50]. The main performance constraints involved in MCM routing are *delay, noise and manufacturability* constraints. Delay constraints are converted to netlength constraints in most cases and require consideration to ensure the the MCM functions properly at the desired clock frequency. Noise constraints are

converted to path separation and parallelization constraints, and they require consideration to avoid unwanted logic switchings. In [41] the author describes in detail the fabrication constraints that should be considered for MCM routing.

As the complexity of the MCM routing increases more computation time will be required as the solution space of the combinatorial optimization problem becomes larger. A distributed computing environment may be necessary to obtain a good solution in a shorter computing time. A concurrent search of the solution space may be necessary with the help of different machines used to configure a PVM (Parallel Virtual Machine) [31]. Better time heuristics can be used to obtain a better quality solution in a PVM environment or the same quality of the solution can be obtained in a much shorter execution time. When a randomized search algorithm like the simulated annealing is used, most of the computation time is spent in calculating the cost function. The cost function calculation can be computed in a set of parallel machines which constitute the PVM. The PVM environment works in a *master - slave* mode. The optimization part is done in the *master* mode using the main host and the cost function calculation goes on in parallel using the *slave* mode. Near-linear speedups are possible using this approach especially for designs having thousands of nets [13].

A variety of routing approaches have been developed in recent years and are described next.

4.2 Maze Routing

The most commonly used MCM routing approach is *maze routing* [42, 13]. This algorithm, which was originally developed by Lee [32], is very simple to use conceptually but suffers from many inadequacies since the quality of the maze routing solution is very sensitive to the ordering of the nets. There is no effective algorithm for determining a good ordering of the nets in general

and a large number of vias are required for routing the nets towards the end even though there are a large number of signal layers. There is also the requirement for large memory to run the algorithm. It is impossible to consider performance constraints such as delay, noise and fabrication constraints in the maze routing algorithm. The high memory requirement for solving large problems as well as the large execution times have made the maze router highly unsuitable for most MCM applications. However, the maze routing algorithm is the backbone of many commercial routers that are available for MCM designs.

4.3 Multiple-Stage Routing

In multi-stage routing, the problem is decomposed into several smaller subproblems. The whole routing cycle is broken up into *pin redistribution*, *layer assignment* and *detailed routing*. These steps were explained in detail earlier in Section 2. Pins on the die layer (top layer) are first redistributed evenly with sufficient spacing in such a way that connections between nets and pins can be done without any design rule violation on signal distribution layers. Then layer assignment is performed so that each net in an x-y pair of layers are assigned subject to the feasibility of routing the nets on a global routing grid in each plane pair. The problem of layer assignment is NP-complete and is discussed in detail in [25, 42, 51, 26, 19]. After layer assignment, the next step is to route the nets using the signal distribution layers. This step, which is known as *detailed routing* depends on the layer assignment step. The approach may use only one single layer or a layer pair. Detailed routing which has been presented earlier in this paper is discussed further in [28].

4.4 Integrated Pin Redistribution and Routing

In this approach instead of distributing the pins before routing, the algorithm redistributes the pins with routing in each layer. After routing on one layer the terminals of the unrouted nets are propagated to the next layer. This concept was developed by Khoo and Cong [9, 30] as follows.

In [9] Khoo and Cong present an integrated routing approach called SLICE for MCM routing. Instead of the usual approach of distributing pins prior to routing, the SLICE router redistributes pins along with routing in each layer. This algorithm follows a layer-by-layer planar routing approach. The algorithm tries to connect as many nets as possible in each layer and the nets which cannot be completely routed in a particular layer are partially routed and then taken over to the next layer after scanning the routing region from left to right. After completing the planar routing in a layer the terminals of the unconnected nets are distributed so that they can be propagated to the next layer without causing local congestion.

In [30] Khoo and Cong have described a general area multi-layer router for MCMs called “V4R” since it uses no more than 4 vias for routing each of the two-terminal nets and no more than $4(k-1)$ vias for each k terminal net. The routing approach is somewhat similar to that of SLICE but V4R operates on x-y plane pairs instead of considering a layer at a time after the whole routing grid has been *sliced* into a number of layers. There is no topological routing step in V4R unlike other MCM routers. The physical routing is generated directly in V4R. The 2-D routing grid is divided into *vertical channels* by vertical columns. A vertical column is defined by a grid line that contains at least one net terminal. Similarly the *horizontal channels* are defined. Every net is routed using one of the two types of routing topologies: Type-1 topologies which have three vertical segments and two horizontal segments and Type-2 topologies which have three horizontal segments and two vertical segments. In each column of the grid, V4R

executes the following steps: *horizontal track assignment of the right terminals, horizontal track assignment of the left terminals, routing in the vertical channel and extension of the routing to the next column*. V4R is able to achieve significant improvements over a 3-D maze router. For example V4R uses 44 % fewer vias and 2 % less wirelength compared to a maze router [30]. V4R has a much faster execution time compared to the maze router and is also very suitable for the increasingly dense MCM designs of today.

A new approach to multi-layer routing for MCMs with pin redistribution is given in [20] and [19]. This router controls various design constraints including coupling between vias and signal lines and discontinuities such as vias and bends. The router redistributes pins or pre-wired subnets uniformly over the MCM substrate using pin redistribution layers. Pin redistribution is very important in MCM design since it not only provides a global distribution for the pins congested in the chip site over the chip layer so as to ease the future routing difficulty, but also reduces the capacitive coupling between vias induced by many layers (up to 63 layers) by separating the pins far apart. The ultimate goal of the problem is to minimize the number of layers required to redistribute the entire set. For signal distribution, a mixed version of *single-layer routing* and *x-y plane-pair routing* techniques was proposed to establish a tradeoff between *circuit performance* and *design objective* instead of emphasizing area minimization only. The strategy supports all of the via types such as stacked (blind) vias and staircase vias. One strategy is to apply single-layer routing iteratively until $\alpha\%$ of the nets are routed, then route the remaining $(100 - \alpha)\%$ nets by x-y plane-pair routing process. This provides the designer with a tradeoff (e.g., between the number of layers and total number of vias) and shows the versatility of the proposed techniques.

4.5 Gridless MCM Router based on Rubber-band Sketches

Grid-based routing schemes are not suitable for large dense MCM designs like the MCM-D [42]. SURF software [11] employs a rubber-band sketch routing approach without an underlying grid. The rubber-band routing approach for an MCM consists of two stages: the creation of a multi-layer rubber-band sketch followed by a conversion of the rubber-band sketch to a physical routing. It uses hierarchical top down partitioning approach to perform global routing of all nets together. The layer assignment is performed during the partitioning process to generate a routing that uses less number of vias and is not restricted to one layer one direction condition. The inputs to this router are a set of terminals, a set of obstacles and a set of wiring rules. The global routing approach followed by the authors of SURF uses two principles from the area of artificial intelligence mainly known as *the least commitment principle* and the notion of *maximal use of information*. The local routing is done one net at a time within the limits of the bin. Both the global router as well as the local router rely heavily on the data structure built on the constrained Delaunay triangulation. SURF generates a topological specification for flexible rubber-band routing. This approach is used for MCM routing across entire MCM substrates that have no channels. The SURF router builds on previous approaches in more than one way. Flexible bin interface specifications allow the global routing to be adjusted as more detailed information is available. By doing the layer assignment during the partitioning step and not restricting the results to the *one-layer-one-direction* constraint, fewer vias are used. This approach to routing represents a good balance between the need to make global decisions and satisfying the constraints imposed by the more detailed local levels.

4.6 Performance-Driven Global Routing

This approach has become one of the most important approaches to MCM routing recently since the role of interconnect in determining the performance of high-speed digital systems has been steadily increasing. It is estimated that interconnect delays account for 50% of the cycle time today, and this number will soon increase to 80%. This is especially true at the MCM level, since the wire-lengths are longer, and the off-chip drivers are bigger and contribute less to the overall delay. Thus, it is very important to consider interconnect delay in MCM routing.

In simplified terms, the delay of an interconnection net is a function of its topology, wire width and metal layer. Ideally, a performance-driven routing algorithm should be able to optimize all three of these simultaneously to achieve maximum performance. However, this is difficult to do in practice because of other constraints such as routability and problem complexity. Typically, the MCM routing problem is solved in three steps: (1) A global routing step determines the topology of each net on a coarse grid. Wire widths may be adjusted to minimize delays after determining the topology; (2) A layer assignment step assigns nets, or segments of nets, to different layers to optimize routability. Sometimes, layer assignment algorithms also consider crosstalk; (3) A detailed area router completes the routing on each layer, following the topology and width suggestions of the global routing.

This section considers the global routing step, and some aspects of the layer assignment problem. Before discussing the algorithms, it is helpful to gain an idea of the models used for the delay of an interconnection.

Delay Models

VLSI interconnect delay models either use the *lumped capacitor* delay model or the *RC tree*

model. The traditional “lumped capacitance” model for an interconnect, in which the delay of a net is proportional to its length, is not accurate enough at the MCM level. Since the effective resistance of off-chip drivers is comparable to the resistance of the interconnect wires, the wire resistance significantly affects the delay and cannot be ignored. Also the delay in a multi-terminal net is not only proportional to the netlength but also to the net topology. Hence we can say that there is a need for more accurate and elaborate delay models for MCMs compared to general VLSI design. The next few paragraphs outline this in greater detail. For a detailed description of the interconnect delay models for MCMs please refer to [42].

For reasonable accuracy, interconnects at the MCM level must be modeled as distributed RLC trees. In this model, each wire segment in the tree is represented by a distributed-RLC section, which consists of a distributed resistance and inductance in series, and a distributed capacitance in parallel to ground. For delay estimation purposes, each distributed section is typically approximated by splitting it into several lumped RLC sections. A first order model for the delay of an RLC tree is the Elmore model [14]. This model ignores the inductance component, and also ignores higher order effects such as “resistance shielding”. However, it has been demonstrated in [3] that this delay model has high “fidelity”, i.e., optimization done based on this model tends to lead to high-performance routes. Figure 2 shows the different delay models used. In [42] the author has developed a second order delay model for multi-terminal interconnects which demonstrates the effect of line inductance on second order delay models.

Higher-order models for interconnect delay can be computed efficiently using techniques such as AWE [37]. Programs such as RICE [38] are efficient enough to be used inside optimization loops.

Topology Optimization

When interconnect resistance becomes significant compared to the effective resistance of the driver, the interconnect delay of a multi-terminal net becomes a function of the topology of the routing. One of the first algorithms which attempted to optimize the topology for delay was presented in [34]. The algorithm iteratively constructs a tree for a multi-terminal net, by starting from a partially constructed tree (which is initially just the driver node), and connecting it to a sink node. The connection path is found using the “A” search algorithm, which is guided by a benefit value on each vertex in the routing graph based on estimates of the Elmore delay to all sinks in the tree, and the total wire length of the tree. An advantage of this algorithm is that it can be used on arbitrary routing graphs, so it can be used on channel graphs, or for area routing in the presence of obstacles. Many other performance-driven algorithms are restricted to the manhattan plane.

Another class of algorithms attempts to minimize Elmore delays directly during construction of the spanning tree for the net. The *Elmore Routing Tree (ERT)* and *Steiner Elmore Routing Tree (SERT)* algorithms [4] construct spanning and Steiner trees, respectively, using a greedy algorithm to minimize Elmore delay.

Using a slightly different delay model called the *dominant time-constant* model, a near-optimal algorithm for minimum-delay routing is presented in [10]. The algorithm is based on the concept of *Arborescence-trees*, which have the property that every driver-sink path is a minimum length path. An A-tree, in general, may have longer wire length than a minimum Steiner tree, but reduces delays when wire resistance is significant. It is shown in [10] that minimizing delay is equivalent to finding an A-tree with minimum wire length. Figure 3 shows the *A-trees* vs the Steiner tree.

An algorithm for finding near-optimal solutions is presented, and experimental results demonstrate delay reductions of up to 40% based on typical MCM parameters. The A-tree algorithm can also be extended to handle obstacles in the Manhattan plane.

Some performance-oriented algorithms do not directly minimize delay, but instead focus on objectives related to delay. For example, close examination of the Elmore delay model reveals that delay is related to the total wire length as well as the square of the source-sink path lengths. This observation is used in [8] to derive a *Bounded Radius Bounded Cost (BRBC)* algorithm, which takes a user-defined parameter ϵ and constructs a tree whose total wire length is at most $2(1 + 2/\epsilon)$ of the optimal Steiner length, and the longest path is at most $(1 + \epsilon)$ times the maximum source-sink distance. A *performance-oriented minimum rectilinear Steiner tree (POMRST)* algorithm [33] minimizes total wire length subject to constraints on the lengths of individual source-sink paths.

In addition to delay, it is sometimes important to minimize the *skew* of a tree, i.e., the maximum difference between the RC delays at different sink nodes in the tree. This is important in clock trees, since the clock signal should arrive at all clock nodes at the same time for correct operation of a synchronous digital circuit. Clock skew can be a significant part of the total cycle time. An algorithm for constructing a tree with *zero skew*, based on the Elmore delay model, was presented in [48]. The algorithm uses a recursive bottom-up strategy, building up a zero-skew tree (ZST) by recursively merging zero-skew subtrees. Figure 4 shows a zero skew tree construction. The original ZST algorithm has undergone improvements to minimize wire length [47] and to guarantee planarity of the resulting tree [27].

Wire sizing

When choosing the wire width of the route for a net, a careful tradeoff has to be made. If

minimum design rules are used for a long wire, the wire resistance will cause large delays. On the other hand, the wire cannot be made arbitrarily wide, since its capacitance will then dominate the delay. The best situation occurs when the wire width is allowed to vary in different parts of the net, being wider near the driver, and narrower near the sinks. This approach, known as *tapering* or *wire sizing*, can reduce RC delays significantly, while simultaneously achieving reduction in wire area (and hence power dissipation).

An optimal algorithm for wire sizing under the dominant time constant model is presented in [10]. A greedy algorithm is also presented in the same paper, which is much more efficient. A combination of the two algorithms produces an efficient optimal algorithm, which can reduce RC delays by as much as 50%.

For the Elmore delay model, a wire sizing algorithm is presented in [40]. A significant feature of this algorithm is that instead of simply minimizing the delay, it allows the user to specify delay constraints at the sink nodes. The algorithm then constructs a wire sizing solution using a sensitivity-based approach, such that the delay constraints are met and the wire area is minimized. This results in significantly better *engineering solutions*, since it is usually not important to make the delay smaller than the constraint, and attempting to do so causes the wire area to increase rapidly. By keeping the delay targets just 15% over the minimum delay, area savings of as much as 46% are observed.

Wire sizing can also be used to solve the skew minimization problem, using an approach similar to the ZST construction. Zero-skew algorithms based on wire length adjustments tend to generate solutions which are very sensitive to process variations: although the tree may have nominally zero skew, the actual skew due to process variations may be large. The algorithm of [35] uses wire sizing to achieve a *reliable* minimum skew solution, by varying wire widths.

4.7 Crosstalk Minimization Routing

Crosstalk refers to the parasitic coupling between neighboring wires due to the mutual inductance and capacitance effects. Crosstalk can increase delays or cause glitches which can cause incorrect operation. Several methods for controlling crosstalk have been proposed, including layer assignment, length control, variable spacing and wire sizing. The author in [13] has developed a crosstalk minimization router. It is claimed that the router developed in [13] not only minimizes the netlength used for routing but also places the wires that carry high frequency signals far apart. This reduces the electromagnetic interaction among the routed wires and thus reduces the *crosstalk*. The number of vias used in the router is controlled by a parameter known as the via control parameter. The input to the router is a *netlist* obtained after the placement of dies on an MCM substrate. Associated with each net is an integer value that represents the *switching activity* (frequency) of the signal carried by the net. Multi-terminal nets are decomposed into two-terminal nets and the router generates orthogonal routing. The routing is carried out by scanning the routing region horizontally or vertically from one end of the substrate to another. Each scan will route as many nets as possible in a pair (horizontal and vertical) of layers.

A layer assignment algorithm for crosstalk minimization is presented in [21], based on an efficient algorithm for maxcut k -color partitioning. Crosstalk minimization algorithms for channel and switchbox routing are presented in [29] and [45], respectively. These ideas could be adapted to MCM routing.

4.8 Three-Dimensional Routing

In [36] the authors have presented a new approach to MCM routing where they present the three-dimensional routing problem as smaller three-dimensional problems to achieve the best utilization of the three-dimensional routing space. The first phase of this approach involves routing distribution in the two-dimensional X-Y plane as well as in the Z direction. This approach ensures uniform routing over the MCM substrate and also satisfies the netlength constraints and manufacturing constraints. The algorithm is designed to take care of the following performance constraints: manufacturability, net-length, net-separation and via minimization. Manufacturability constraint should be such that the yield is maximum. When considering signal delays it is necessary to consider the delays in critical nets. The *netlength constraint* can be represented as,

$$l_i < \alpha_i \text{ for } n_i \in N$$

where, n_i represents each individual net, l_i represents the corresponding longest path from the source of n_i to the sink of n_i and α_i represents the length constraint

Net separation constraint primarily takes care of the crosstalk. The crosstalk between two lines can be minimized by ensuring that for each pair of nets the minimum separation is greater than a certain specified limit. The *via constraint* is introduced to minimize the maximum number of stacked vias for a net and satisfy the fabrication requirement. The via constraint is stated as,

$$v_i < \gamma \text{ for } n_i \in N$$

where the maximum number of stacked vias for a net is represented as γ and the number of stacked vias of net n_i at any given point is v_i . MCM routing is carried out in a three-dimensional routing space using a recursive formulation. The routing is completed in several phases which are

tiling, off tile routing, two-dimensional routing distribution, Z - dimension routing distribution, terminal assignment and tower routing. Tiling involves splitting the substrate area into smaller regions such that the denser regions are partitioned more compared to the less dense regions. The main function of the *two-dimensional routing distribution* is to see that the *manufacturability constraint* is satisfied. At the end of this phase the number of routing layers based on the routing congestion is estimated. The length of each net in the X-Y direction is also estimated. The objective of the *Z-dimension routing distribution* phase is to uniformly reduce congestion in the Z-direction. The terminal assignment phase is carried out by bipartitioning the substrate recursively.

The researchers who have developed this approach claim that the general objective of performance optimization using as few layers as possible has been satisfied using this three-dimensional approach.

4.9 Comparison of the Different Routing Algorithms

A lot of different routing algorithms for multi-layer MCM routing are available, each having some pros and cons. A comparative evaluation of the different algorithms is given in Table 2 and Table 3.

5 Conclusions and Scope for Future Work

A lot of work has already been done by different authors in the area of *placement and routing*. Most of the work in the area of MCM placement has focused on netlength minimization alone though some of the researchers have also considered the problem of proper heat distribution when placing the dies on the substrate. An automatic placement tool that considers both the

netlength minimization as well as the thermal criteria has also been developed. However there is a need to consider the wire resistance in delay estimation as the net delay is not just a function of the wirelength. More accurate algorithms need to be developed that would extend the concept of resistance-driven placement to RLC driven placement, to further reduce the interchip signal delays. Thermal constraints also have to be taken into account more carefully. Also it is no longer sufficient to do the substrate and IC placement in multi-chip modules individually in a standalone fashion [16]. Optimizing an MCM system requires close attention to the physical designs of the ICs and the substrate simultaneously. This type of approach will relax the routing constraints which are usually placed on the substrate when the IC designs have been completed before the substrate.

It is becoming clear to the IC and MCM design community that interconnects are no longer just “parasitics” - they are an integral part of the circuit design. Taking interconnects into account during circuit design can significantly improve system performance. To this end, new research, such as [18], is focusing on the problem of *simultaneous driver and wire sizing*. [18] presents efficient optimal algorithms for simultaneous driver and wire sizing for delay minimization, or delay and power minimization. Figure 5 shows a tapered interconnect.

Since clock distribution trees on an MCM can be very large, and heavily loaded, it is very difficult to drive them with a single driver. Typically, *buffers* are introduced at various nodes in the tree, to avoid having a single very large driver, and to keep the slope of the received waveform reasonably high. Such *multi-stage clock trees* introduce some interesting new optimization problems, such as finding the optimal locations for the buffers, minimizing the number of buffers, finding optimal sizes for the buffers etc. An algorithm for minimizing the number of buffers in a given clock tree subject to a clock period constraint is presented in [46]. The basic

algorithm is extended to handle upper bound constraints on the skew, and allow area and delay tradeoffs. Figure 6 shows a multi-stage clock tree distribution.

The disadvantage of constructing a matching-based hierarchical zero-skew clock tree is that the wires dictated by the matching edges in higher level of hierarchy are relatively long and they may introduce both a severe crosstalk noise due to buffer congestion of the generated tree topology. The local congestion problem can be eliminated by distributing the buffers over the plane with minimum impact on wirelength. By reducing the congestion of buffers, crosstalk delay will also be significantly reduced. Thus, the authors in [24] investigated the problem of reducing congestion, wirelength and clock skew during the growth of the clock tree. To take the three performance constraints into account simultaneously, the buffer distribution was formulated as a minimum length degree distributed spanning tree problem. An efficient solution to the problem is proposed in [24]. The proposed algorithm can be applied to MCM clock net routing. H-tree is preferable for clock distribution in the transmission line mode, since reflections and crosstalk can be minimized [2]. Thus, a clock tree routing scheme for hierarchical packaging system (e.g., MCM) would be as follows. An H-tree is used for inter-chip clock routing generating a set of clock sub-sources, each of which is inside each chip. Whereas pins in each chip are interconnected from the sub-source only inside the routing subregion using the proposed clock tree construction scheme.

The past few years have generated a large body of important results on interconnect optimization. However, there are still many areas which are unexplored. For example, performance-driven tree construction and wire sizing algorithms do not handle situations where a net has more than one driver. This frequently occurs on global signals like data and control busses, where performance is critical. Other important areas requiring further work are more accurate

modeling of transmission line effects and driver non-linearities. Currently, most routing algorithms use distributed RC delay models which do not take into account the inductive behavior of MCM interconnects. Furthermore, the driver is usually modeled as a single linear resistor, which can be a very inaccurate model. Finally, the various performance-optimization approaches - topology optimization, wire sizing and layer assignment - need to be unified into a single algorithmic framework and made to handle additional physical constraints such as congestion and routability.

Recently Field Programmable Multichip Modules (FPMCM) have been developed by researchers in [23], [22]. MCM technology has lead to a great improvement in the performance and yield of FPGAs by bringing about an integration of MCM and FPGA technology. This new technology is called FPMCM (Field Programmable Multichip Module). Most of these FPMCMs use area I/O flip chip type of technology instead of the regular wire bond peripheral pads. This necessitates the need for a whole new set of tools for area pad assignment, placement and routing of FPMCMs.

6 Acknowledgments

The authors would like to express their special thanks to Prof. Jun Dong Cho from Sung Kyun Kwan University, Korea for his very valuable suggestions while writing this paper. We appreciate his help and encouragement. The authors would also like to express their thanks to Dr. M. Sriram of INTEL Corporation, Santa Clara for his very helpful suggestions during the course of writing this paper. His work in the area of MCM physical design has helped in a big way to identify some of the research done in the area of MCM placement and routing.

References

- [1] K. Chaudhury A. Srinivasan and E. S. Kuh. " RITUAL : A Performance Driven Placement Algorithm ". In *IEEE Transactions on Circuits and Systems*, 39(11):825-840, November 1992.
- [2] H.B. Bakoglu. " *Circuits, Interconnections and Packaging for VLSI*". Addison - Wesley, 1990.
- [3] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins. " Fidelity and Near-Optimality of Elmore-Based Routing Constructions ". Technical Report CS-93-14, Dept. of CS, Univ. of Virginia, Charlottesville, 1993.
- [4] K. D. Boese, A. B. Kahng, and G. Robins. " High-Performance Routing Trees with Identified Critical Sinks ". In *Proceedings of Design and Automation Conference*, pages 182-187, June 1993.
- [5] M. Burstein. " A Non Placement-Routing Approach to Automation of VLSI Layout Design ". In *Proc. Intl. Symposium of Circuits*, pages 234-244, 1989.
- [6] C. Harvatis, S. Mehrotra, P. Franzon. " Topological Description and Interconnect Delay Modeling of Performance-Driven Partitioning and Placement of MCM Design ". Technical Report NCSU-ERL-94-17, Dept. of Electrical Engineering, North Carolina State University, 1994.
- [7] J. Cohoon and W. Paris. " Genetic Placement ". In *IEEE Transactions on Computer-Aided Design of Circuits and Systems vol. CAD-6,no. 6*, page 956-964, November 1987.

- [8] J. Cong, A. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong. " Provably Good Performance-Driven Global Routing ". *TCAD*, 11(6):739–752, June 1992.
- [9] J. Cong and K.Y. Khoo. " An fast multilayer general area router for MCM designs ". In *IEEE Transactions on Circuits and systems*, pages 841–851, November 1992.
- [10] J. Cong, K.-S. Leung, and D. Zhou. " Performance-Driven Interconnect Proceedings of Design Based on Distributed RC Delay Model ". Technical Report CSD-920043, UCLA, October 1992.
- [11] W.W. Dai. " Topological Routing in SURF: Generating a Rubberband Sketch ". In *Proc. IEEE Design Automation Conference*, pages 39–48, 1991.
- [12] P. H. Dehkordi and D. W. Bouldin. " Design for Packageability - Early Considerations of Packaging from a VLSI Designer's Viewpoint ". In *IEEE Computer*, pages 76–81, April 1993.
- [13] G. Devaraj. " *Distributed Placement and Crosstalk Driven Router for Multichip Modules* ". MS Thesis, University of Cincinnati, 1994.
- [14] W.C. Elmore. " Transient Response of Damped Linear Response with Particular regard to Wideband Amplifiers ". In *Journal of Applied Physics*, pages 55–63, 1948.
- [15] H. Esbenson and P. Mazumder. " SAGA - A Unification of Genetic Algorithm with Simulated Annealing and its Application to Macrocell placement ". In *Proceedings of 7th International Conference on VLSI Design, Calcutta*, pages 211–214, January 1994.
- [16] Cadence Design Systems Inc. " A Vision for Multichip Module Design in the Nineties ". Technical report, Cadence Design Systems Inc., Santa Clara, CA, 1993.

- [17] C. King Iosupovici, A and M.A. Breuer. " A Module Interchange Placement Machine ". In *Proceedings of Custom Integrated Circuit Conference*, May 1984.
- [18] J. Cong and C.K. Koh. " Simultaneous Driver and Wire Sizing for Performance and Power Optimization ". In *ICCAD*, pages 206-212, November 1994.
- [19] J. D. Cho and M. Sarrafzadeh. " The Pin Redistribution Problem in Multichip Modules ". In *Mathematical Programming, North Holland - Special Issue on Applications of Discrete Programming in Computer Science*, pages 297-330, 1994.
- [20] K. F. Liao J. D. Cho, S. Raje and M. Sarrafzadeh. " A New Multilayer Routing System for High Performance MCM's ". In *IEEE Transactions on Circuits and Systems, Vol 41, No 4*, pages 253-255, April 1994.
- [21] J. D. Cho, S. Raje and M. Sarrafzadeh and M. Sriram and S. M. Kang. " Crosstalk-Minimum Layer Assignment ". In *IEEE Custom Integrated Circuits Conference*, May 1993.
- [22] R. Ramirez J. Darnauer, P. Garraay and W. Dai. " A Field Programmable Multichip Module ". In *IEEE Workshop on FPFA for Custom Computing Machines*, pages 64-67, 1994.
- [23] V. Maheshwari J. Darnauer, T. Isshiki and W. Dai. " FPMCM - An Integration of FPGA and MCM technology ". In *IEEE Multichip Module Conference, MCMC - 1995*, pages 50-55, 1995.
- [24] J.D. Cho and Majid Sarrafzadeh. " A Buffer Distribution Algorithm for High-Performance Clock net Optimization ". In *IEEE Transactions on VLSI Systems, Vol.3 No. 1*, March 1995.

- [25] G. Vijayan J.M. Ho, M. Sarrafzadeh and C. K. Wong. " Layer Assignment for Multichip Modules ". In *IEEE Transactions on Computer Aided Design*, 9(12), pages 1272–1277, 1990.
- [26] G. Vijayan J.M. Ho, M. Sarrafzadeh and C.K. Wong. " Layer Assignment for Multichip Modules ". In *IEEE Transactions on Computer Aided Design*, pages 1272–1277, December 1990.
- [27] A. B. Kahng and C. W. A. Tsao. " Low-Cost Single Layer Clock Trees With Exact Zero Elmore Delay Skew ". In *ICCAD*, pages 213–218, November 1994.
- [28] M. Sarrafzadeh K.F. Liao and C. K. Wong. " Single Layer Global Routing ". In *Proceedings of Fourth Annual IEEE International ASIC Conference and Exhibit*, pages 14–4.1 – 14–4.4, 1991.
- [29] D. Kirkpatrick and A. L. Sangiovanni-Vincentelli. "Techniques for Crosstalk Avoidance in the Physical Proceedings of Design of High-Performance Digital Systems ". In *Proceedings of ICCAD*, pages 616–621, November 1994.
- [30] K.Y. Khoo and Jason Cong. " An Efficient Multilayer MCM Router Based on 4-via Routing ". In *Proc. IEEE/ACM Design Automation Conference*, pages 590–595, June 1993.
- [31] Oak Ridge National Laboratory. " PVM - 3 Users guide and Reference Manual ". Technical report, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1991.
- [32] C.Y. Lee. " An Algorithm for Path Connection and its Applications ". In *IRE Transactions on Electronic Computers*, 1961.

- [33] A. Lim, S.W. Cheng, and C.T. Wu. " Performance Oriented Rectilinear Steiner Trees ".
In *Proceedings of Design and Automation Conference*, pages 171–176, June 1993.
- [34] S. Prasitjutrakul and W. Kubitz. " A Timing-Driven Global Router for Custom Chip Design ". In *ICCAD*, pages 48–51, 1990.
- [35] S. Pullela, N. Menezes, and L. T. Pillage. " Reliable Non-Zero Skew Clock Trees Using Wire Width Optimization ". In *Proceedings of Design and Automation Conference*, pages 165–170, June 1993.
- [36] Q. Yu, J.L. Lai, S. Badida and N. Sherwani. " A Performance Driven Three Dimensional Approach to MCM routing ". Technical report, Dept. of CS, Western Michigan University, September 1993.
- [37] V. Raghavan, R. A. Rohrer, L. T. Pillage, J. Y. Lee, J. E. Bracken, and M. M. Alaybeyi. " AWE-Inspired ". In *Custom Integrated Circuits Conference*, pages 18.1.1–18.1.8, May 1993.
- [38] C. L. Ratzlaff, N. Gopal, and L. T. Pillage. " RICE: Rapid Interconnect Circuit Evaluator ". In *Proceedings of Design and Automation Conference*, pages 555–560, June 1991.
- [39] P. A. Sandborn and H. Moreno. " *Conceptual Design of Multichip Modules and Systems* ". Kluwer Academic Publishers, 1993.
- [40] Sachin S. Sapatnekar. " RC Interconnect Optimization Under the Elmore Delay Model ". In *Proceedings of Design Automation Conference*, pages 387–391, June 1994.
- [41] N. Sherwani. " *Algorithms for VLSI Physical Design Automation* ". Kluwer Academic Publishers, 1993.

- [42] M. Sriram and S.M. Kang. “ *Physical Design of Multichip Modules* ”. Kluwer Academic Publishers, 1993.
- [43] P.R. Suaris and G. Kedem. “ A Quadrisection Based Combined Place and Route scheme for Standard Cells ”. In *IEEE Transactions on Circuits and Systems*, 8(3), pages 234–244, 1989.
- [44] P.M. Vaidya T. Gao and C.L. Liu. “ A New Performance driven Placement ”. In *Dig. Tech. Papers, Intl. Conf on Computer Aided Design*, pages 44–47, 1991.
- [45] T. Gao and C. L. Liu. “Minimum Crosstalk Switchbox Routing”. In *Proceedings of ICCAD*, pages 610–615, November 1994.
- [46] G. E. Tellez and M. Sarrafzadeh. “ Clock Period Constrained Minimal Buffer Insertion in Clock Trees ”. In *Proceedings of ICCAD*, pages 219–225, November 1994.
- [47] Y.C. Hsu T.H. Chao and J.M. Ho. “ Zero Skew Clock Net Routing ”. In *Proceedings of Design and Automation Conference*, pages 518–523, June 1992.
- [48] R.S. Tsay. “ Exact Zero Skew ”. In *ICCAD*, pages 336–339, November 1991.
- [49] Rao R. Tummala. “ Multichip packaging - A Tutorial ”. In *Proceedings of IEEE*, pages 1924–1942, 1992.
- [50] R.R. Tummala. “ High Performance Glass-ceramic/copper Multilayer Substrate With Thin Film Redistribution ”. In *IBM Journal of Research and Development*, pages 889–902, 1992.
- [51] R. Vemuri. “ *Genetic Algorithms for Partitioning Placement and Layer Assignment for Multichip Modules* ”. Ph.D Dissertation, University of Cincinnati, 1994.

Biographies

Subhomoy Chattopadhyay Subhomoy was born in Calcutta, India. He obtained his BSEE and MSEE degrees in Electrical Engineering from the University of Calcutta and the Indian Institute of Technology, Madras in India in 1989 and 1991 respectively. He obtained his MS degree in Electrical and Computer engineering with specialization in VLSI and Multichip module design from University of Tennessee, Knoxville, TN, USA in 1995. He was also working part time on an US ARPA funded project on Multichip modules at University of Tennessee. He is presently employed at INTEL Corporation, Folsom, California. His area of interests include VLSI and Multichip module physical design and automation.

Donald W. Bouldin Dr. Don Bouldin is an IBM Professor of Electrical and Computer Engineering at the University of Tennessee, Knoxville. He received the B.E. (Electrical Engineering) from Vanderbilt University in 1967, the M.S. (Electrical Engineering) from Georgia Institute of Technology in 1968, and the Ph.D from Vanderbilt University, Nashville in 1975. Dr. Bouldin has authored more than 120 publications and been the Principal Investigator for nearly four million dollars of sponsored research, primarily with the nearby Oak Ridge National Laboratory, NSF and ARPA. Prof. Bouldin served two years as Chairman of the Technical Committee on VLSI for the IEEE Computer Society. He was instrumental in gaining IEEE approval for the IEEE Transactions on VLSI Systems and currently serves as its Editor-in-Chief.

Peyman H. Dehkordi Dr. Peyman Dehkordi is a Research Assistant Professor in Electrical and Computer Engineering at the University of Tennessee in Knoxville. He received his B.S., M.S., and Ph.D. in Electrical Engineering from University of Tennessee in Knoxville in

1981, 1985, and 1992. During 1982-1993, he developed real-time image-processing inspection systems for national and international markets. He has been the technical lead for over \$3 million in research and development for ARPA, SBIR, and private industries since 1992. He holds two US and one international patents. He is currently responsible for the technical leadership for the "Design for Packageability" ARPA project at University of Tennessee exploring design optimization techniques for multichip module designs.

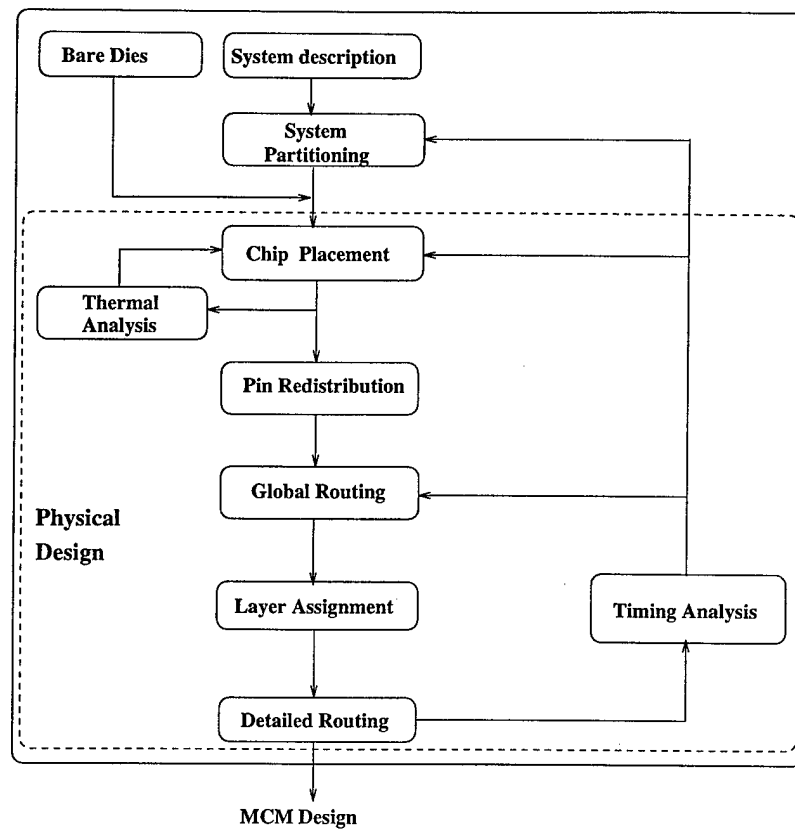


Figure 1: *MCM Physical Design - Flow*

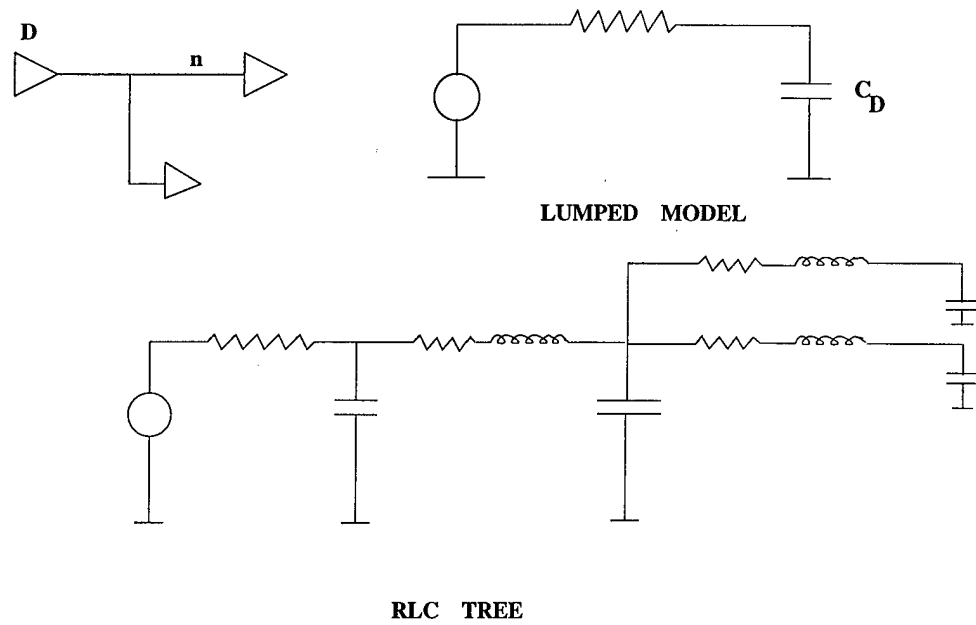


Figure 2: *Some Delay Models*

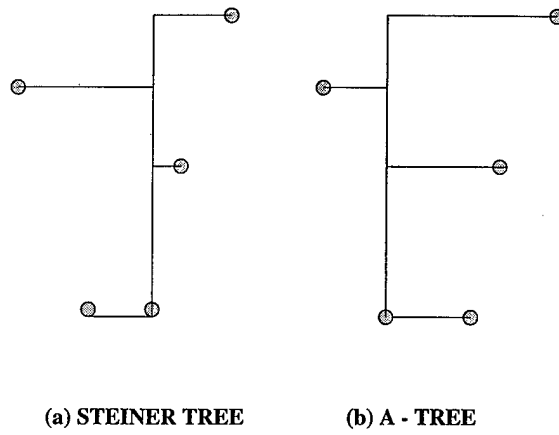


Figure 3: *A - Tree vs Steiner Tree*

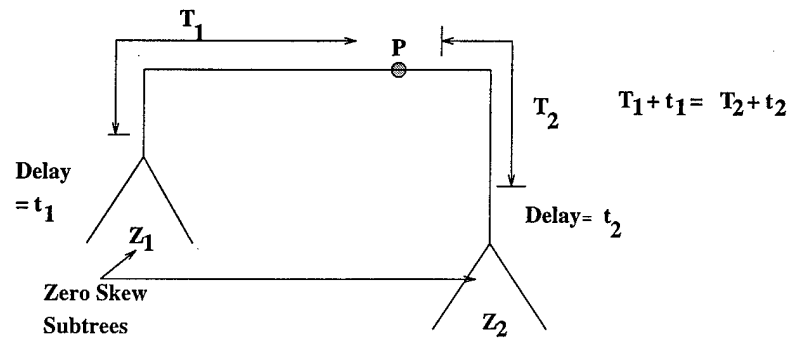


Figure 4: *Zero - Skew Tree Construction*

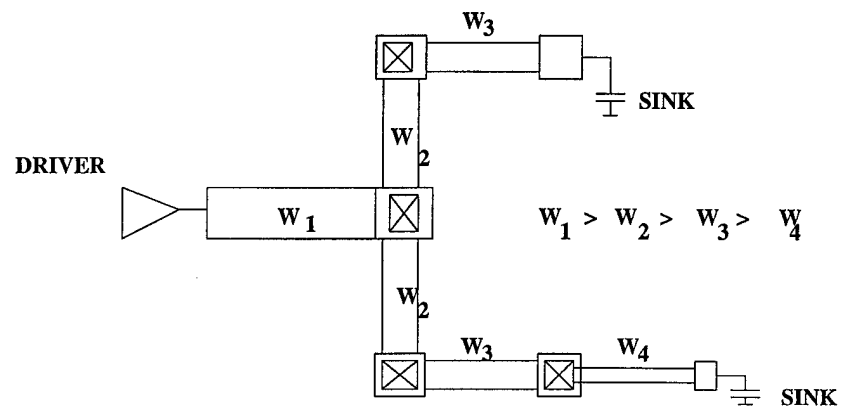
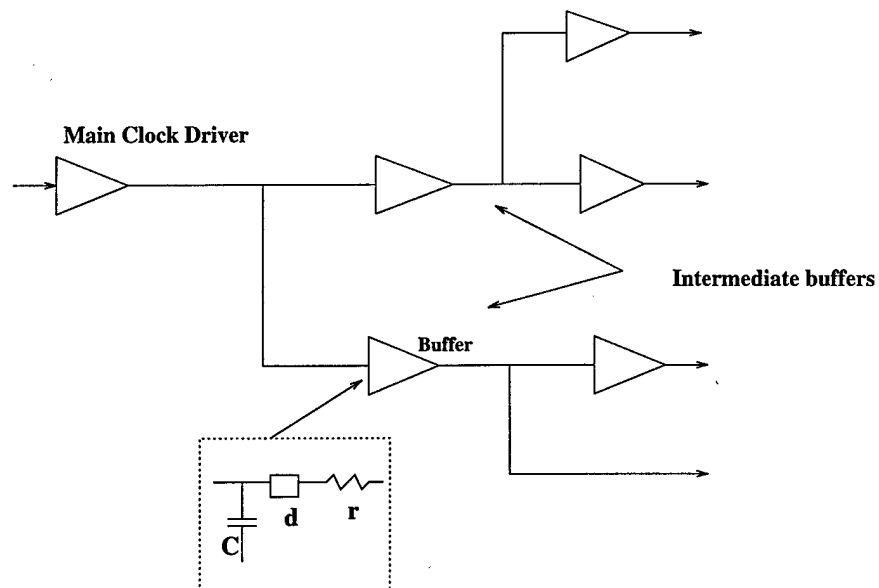


Figure 5: A tapered interconnect



C : Input Capacitance
d : Intrinsic delay
r : Output impedance

Figure 6: A Multi Stage Clock Tree Distribution

Author, Ref, Date	Netlength Minimization	Thermal Constraint	Performance Driven	Remarks
Franzon [6], 1995	Yes	No	Timing-driven	High probability of placements satisfying timing criterion
Vemuri [51],1994	Yes	Yes	No	Genetic algorithm iterative, gives a good solution
Devaraj [13],1994	Yes	No	No	SA-based, no thermal criteria, parallel implementation using PVM
Sriram [42],1993	Yes	No	Resistance- driven	Interchip signal delays reduced

Table 1: Comparison of Different MCM Placement Algorithms

Approach	Reference and Date	Netlength Minimization	Via Minimization	Layer Minimization	Crosstalk Minimization
Maze Router	[32], 1961	Yes	No	No	-
SLICE Router	[9],1991	Yes	Yes	Yes	-
V4R Router	[30],1993	Yes	Yes	Yes	-
Rubber-band Router	[11],1991	Yes	Yes	Yes	Yes
Performance Driven	[42],1993	Yes	Yes	Yes	Yes
Crosstalk Minimization	[13],1994	Yes	Yes	Yes	Yes
3D routing	[36],1993	Yes	Yes	Yes	No

Table 2: Comparison of MCM Routing Algorithms, Part I

Approach	Wiresizing	Runtime	Memory Requirement	Remarks
Maze Router	-	Large	Large	Heavy duty, inefficient
				large memory required
SLICE Router	-	Not large	Not large	Good for moderate sized designs
V4R Router	-	Not large	-	Improved version of SLICE router
Rubber-band Router	Yes	NI	NI	Gridless router, good for high density MCM designs
Performance Driven	Yes	Large	NI	Good and most accurate modeling of delays
Crosstalk Minimization	Yes	Fast	Not Large	Variable via and number of layers
3D routing	-	Not Large driven	Small	Efficient utilization of 3D routing resources

Table 3: Comparison of MCM Routing Algorithms, Part II; NI = no information available

Dehkordi, P., Powell, T., and D. Bouldin,

“Development of a DSP/MCM Subsystem: Assessing Low-Volume, Low-Cost MCM Prototyping for Universities”, *Proceedings of 1996 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, February 5-7, 1996.

Development of a DSP/MCM Subsystem

Assessing Low-volume, Low-cost MCM Prototyping for Universities

Peyman Dehkordi, Tim Powell, Donald Bouldin
University of Tennessee¹

Abstract

This paper discusses the design and development of a general-purpose programmable DSP subsystem packaged in a multichip module. The subsystem contains a 32-bit floating-point programmable DSP processor along with 256 K-byte of SRAM, 128 K-byte of FLASH memory, 10 K-gate FPGA and a 6-channel 12-bit ADC. The complete subsystem is interconnected on a 37 mm by 37 mm MCM-D substrate and packaged in a 320-pin ceramic quad flat pack. The design has been submitted to the MIDAS brokerage service to be fabricated by Micro Module Systems. Our experience shows that low-volume MCM prototyping is achievable and somewhat affordable for universities. The design flow, electrical and thermal analyses, CAD tools, cost and lessons learned are discussed in this paper.

Introduction

Multichip module packaging has been in use by IBM and others for high-volume and/or high-cost products for many years but only recently has a low-volume and low-cost service become available to the design community. A variety of other reasons have also impeded growth including the availability of known good die (KGD) and the expense of sophisticated CAD tools optimized for this purpose. Previous work such as [1] discusses various MCM design techniques in depth; however, real-world MCM design problems are not well documented for the first-time users [2]. There has been a tremendous push from Advanced Research Project Agency (ARPA) for wide-spread usage of this technology for the past few years through their Application Specific Electronic Module (ASEM) program. As a result, MIDAS is now offering MCM services for low-cost prototyping [3]. As a part of our ARPA project, we have exercised the availability of this technology for universities. We have performed this exercise by acquiring the appropriate CAD tools and designing a subsystem using bare dies that were procured in small quantities within the required time period. The final design has been submitted to MIDAS for fabrication.

Design Flow

We decided to implement a design representing a typical application of a MCM including a processor, memory and gate-array. It consists of a complete DSP subsystem based on the Motorola 96002 32-bit floating point DSP processor as shown in Figure 1. The module also includes 256 K-byte of SRAM, 128 K-byte of EEPROM, a 10 K-gate Xilinx 4010 FPGA and a 6-channel 12-bit ADC. The module was designed to exploit the multi-processing capability of the 96002. The FPGA is primarily a hardware pre-processor of incoming data but also provides other generic logic functions required for implementation of the entire subsystem.

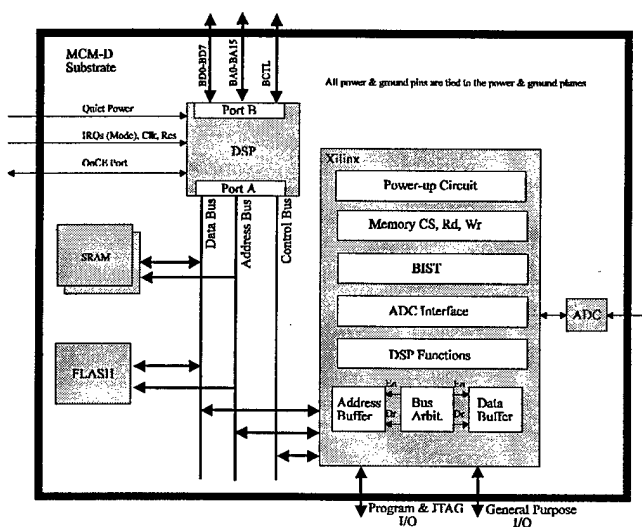


Figure 1. The Block Diagram of the DSP Sub-system.

The design flow is shown in Figure 2 along with the CAD tools used. We had to modify our original plans several times in order to be certain that all of the dies could be acquired within the desired time frame. We found out that it is a good practice to establish a good working relationship with the die supplier to reduce the design cycle as well as die cost and to take advantage of the availability of dies which have already been tooled by the supplier. We

¹ This work was sponsored in part by ARPA under grant DAAH04-94-G-0004

performed an early analysis on the candidate design to make sure the selected components would fit within the various available substrate sizes offered by MIDAS.

We were constrained to select one of a few choices of substrates and packages offered by MIDAS at the time. The substrate selected is a 37 mm x 37 mm 5-layer (1 pad, 2 power, and 2 routing layers, 62 μ pitch) MCM-D based on the MMS *Merged Via D500* process [4]. It utilizes copper/polymide for interconnect/dielectric and an aluminum substrate as a ground plane. The package is a cavity-down, 320-pin ceramic quad flat pack with preassigned power and ground pins and a lead pitch of 0.65 mm.

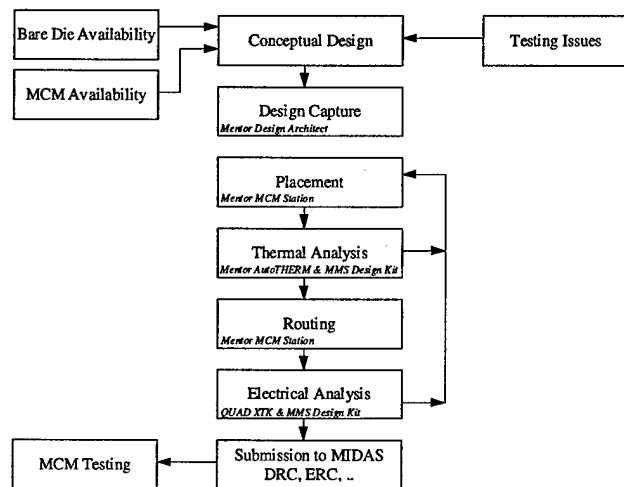


Figure 2. The Design Flow and CAD Tools.

Obtaining the bare dies was probably the most challenging part of this exercise. Traditionally, third party die suppliers have been providing bare dies for the MCM and hybrid markets. Due to increased MCM activities, some of the semiconductor manufacturers (i.e Intel, Motorola) have started selling some of their products in bare die form. For example, Intel offers some of its dies through its *Smart die* program. We have found that there is a long lead time (16 weeks) with these parts and it is necessary to execute a legal agreement between the supplier and the university. This step alone can delay a project by several months in our experience. Another issue was the minimum quantity which suppliers wanted us to buy was sometimes several hundred dies costing several thousand dollars, which exceeded our budget for this low-volume project.

We contacted three die suppliers; however, only Chip Supply was very helpful with our die needs. *Known Good Die* was another issue we faced [5]. A true KGD (fully tested at speed and burned-in) could be very expensive, especially considering that the NRE cost for tooling a new

die may be \$15,000. We ended up purchasing bare dies which had been only wafer probed and visually inspected. All other bare die were donated to us and had undergone similar testing which was acceptable since we will prototyping only five copies of our MCM.

Electrical Analysis

Once the design was finalized and all the components were selected, the MCM physical design was performed using the *Mentor Graphics MCM Station*. The foundry design kit for MMS was obtained via MIDAS and facilitated template generation for the components and their bond-pads on the substrate. This included all the connections to the power and ground layers as well as thermal via generation for each die. Once a good placement of the components was reached (conforming to the electrical and thermal constraints), the critical nets were manually routed and the autorouter was then invoked. Figure 3 shows the final routed layout.

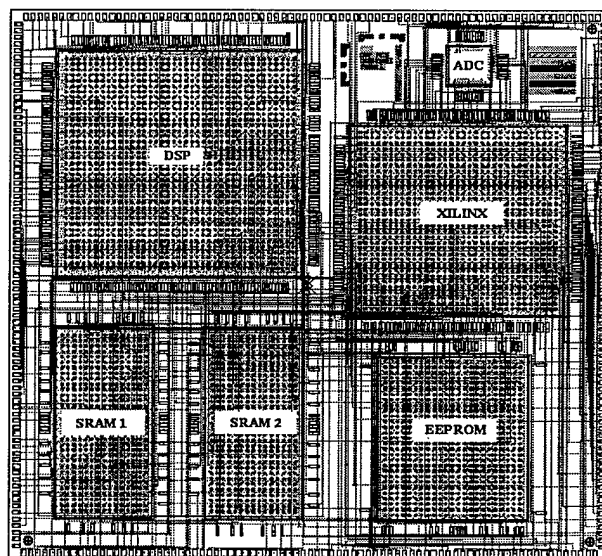


Figure 3. The Physical Layout of the MCM.

None of the components were speed-rated due to the procurement reasons described above. We assume a nominal frequency of operation of about 40 MHz since the DSP die has come from a 40 MHz lot. The entire design was simulated with QUAD's XNS tool which uses a time-domain finite element analysis technique to compute over/under shoot, simultaneous switching noise, rise/fall times, and coupling noises on circuit networks of arbitrary topology. A typical multi-drop net model and its electrical simulation are shown in Figure 4 and 5 respectively.

The maximum delay was found to exist between the DSP and the FPGA on an address line and computed to be 1.8 nS. Approximately 55% of the signals are under 1.0 nS

delay and the longest total net length including all branches was 77.286 mm. The delay distribution on the module is shown in below:

percent paths exceeding	
0 10 20 30 40 50 60 70 80 90 100	
time(nS)	----- ----- ----- ----- ----- ----- ----- ----- ----- -----
0.00	*****
0.17	*****
0.33	*****
0.50	*****
0.67	*****
0.83	*****
1.00	*****
1.17	*****
1.33	***
1.50	**
1.67	*

The maximum crosstalk was also induced on an address line and was simulated to be 214 mV. The crosstalk was only calculated on the plane of the routed trace. The other routing plane which might couple intersecting traces was not considered. The reflections were simulated to be low since the source impedance of the driver added to the line resistance presenting a source-series termination [6]. The simulation data suggests that the limiting speed factor in this design (with the current models) would be the data bus. Almost all data lines exhibited noticeably more slewing than address lines. We were not able to get the I/O models from the die vendors. Various published I/O models from other vendors were used in the simulation. Although the results may not be very accurate (due to I/O buffer assumption), they can outline the trouble nets and general characteristics of the design. A detailed analysis is presented in [7].

Thermal Analysis

MCMs integrate several large VLSI dies consuming several watts within a small area. Thermal analysis of an MCM is very important to ensure a reliable operation. The objective of thermal analysis is to predict the junction temperature of the dies in determining long-term reliability of the components and the module. The thermal path from the die to the heat sink is comprised of the following three paths [4]:

$$\begin{aligned} \text{Path}_1 &= R_{\text{die}} + R_{\text{die epoxy}} + R_{\text{interconnect}} + R_{\text{interconnect epoxy}} \\ \text{Path}_2 &= R_{\text{interconnect epoxy}} \\ \text{Path}_3 &= R_{\text{thermal grease}} + R_{\text{heat sink}} \end{aligned}$$

Thermal vias were placed under all the dies to improve the thermal conductivity path from the dies to the backing plate. Thermal vias were created in such a fashion to allow two traces between them. This resulted in less routing

penalty under the dies due to the thermal vias. The junction temperature distribution is shown in Figure 5 for an ambient temperature of 25° C. The results were generated by the *Mentor Graphic AutoTHERM* using two-dimensional finite-element mesh analysis. The maximum junction temperature was simulated to be 50° C at the DSP which is well within the maximum rating of the dies. The module specification is shown in Table 1.

Cost & Testing Issues

The purpose of this section is to provide the reader with a *rather realistic* cost for this MCM. We have prototyped a total of 5 MCMs which each MCM costing about \$ 2,780. The total cost of the bare die was \$ 1,300, assuming retail prices for the donated die. The cost of the bare dies included the dies and wafer-probe and visual inspections for the *stocked* dies. The substrate cost of \$ 1,480 per MCM included the fabrication and test of the substrate, assembly of the dies onto the substrate and the CQFP package. The cost associated with the final system testing, design time, and the CAD tools have *not* been included in the \$2,780.

Due to their small feature sizes, MCMs cannot easily be probed or reworked [8]. Therefore, testing and fault isolation becomes a challenging area for MCM designs. Complete boundary scan testing was not feasible for this MCM since only the FPGA supported JTAG. The DSP supports a serial emulation port which was connected to the module pins along with the FPGA TAP port for testing purposes. The FPGA has a RAM-based configuration where the configuration may come from outside the module. A portion of the FPGA is configured for multi-processing bus arbitration and is connected to the internal data and address bus on the module. The FPGA can easily be reconfigured to route these lines to the module pins for possible probing and debugging during testing. The DSP has a dual data and address port architecture in which the second port was also connected to the module pins for enhanced multi-processing and testing. Testing procedures are being constructed at the time of this publication.

Conclusion

Multichip modules are becoming a viable choice of packaging for high performance / miniaturized electronics. This technology is no longer considered to be an exotic technology and is being used in telecommunications, consumer electronics and workstations. Low-cost, low-volume prototyping capabilities are a "must" for university related programs as well as small/medium size companies who wish to utilize this technology. We have found out that accessing this technology is becoming a reality for university programs at *almost* low-cost. The MMS design-kit was *very* useful not only with the physical layout but also since it provided a variety of check points and design tips. Availability of the bare dies (and their I/O buffer models),

testing and rework issues will continue to be challenging issues to be considered. We are looking forward to getting the MCM back to confirm the functionality as well as validating the thermal/electrical analysis.

Substrate	37 mm by 37 mm MCM-D
Package	320 pin CQFP
Power consumption	10 W
Power Density	0.73 W/cm ²
Number of dies	6
Number of signal nets	284
Silicon Efficiency	51 %
Average net length	22 mm
Longest net length	77 mm
Total vias	510
Total wire-bonds	930
Module Cost	\$ 2,780 (MCM + dies)

Table I) Module Specification

Acknowledgments

The authors gratefully acknowledge the donation of the bare-dies from Sam Valencia (Motorola), Mitch Richman (Xilinx), and Chuck Britton (Oak Ridge National Lab.). We would also like to thank Jennifer Peltier from MIDAS/ISI for her technical assistance throughout the project.

References

- [1] Sarfaraz, A., Swaminathan, M., Crocker, J., Bhatia, H., and M. Nealon, "Electrical Design of an MCM Package for a Multi-Processor Digital System", *IEEE Transactions on Components, Packaging, and Manufacturing Technology-Part B*, Vol. 18, No. 1, February 1995.
- [2] Ladd, S., and J. Mandry, "Solving Real World MCM Design Problems", *Proceedings of the International Conference and Exhibition on Multichip Modules*, April, 1994.
- [3] Peltier, J., and W. Hansford, "MIDAS: A service to Obtain Low Cost Prototype Quantities of MCM's from Commercial Foundries", *Proceedings of the International Conference on Multichip Modules*, April, 1995.
- [4] MicroModule Systems, Mentor Graphics, "MCM-D Technology Kit, MicroModule Systems MCM-D500 Process", September, 1994.
- [5] Begay, M., Zee, J., Westbrook, G. and B. Williams, "Getting to Know Your MCM Die", *Proceedings of the International Conference and Exhibition on Multichip Modules*, April, 1993.
- [6] Tuckerman, D., "Electrical Design Techniques for MCM-D", *Design Tutorial of IEEE Multichip Modules Conference*, March, 1994.
- [7] Powell, T., "The Development, Design, and Comparison of Printed Circuit Board and MultiChip Module Representations of a DSP Subsystem, *Master Thesis*, University of Tennessee, December 1995.
- [8] Flint, A. and W. Blood, "Board Test in an IC Environment", *Proceedings of the International Conference and Exhibition on Multichip Modules*, April, 1993.

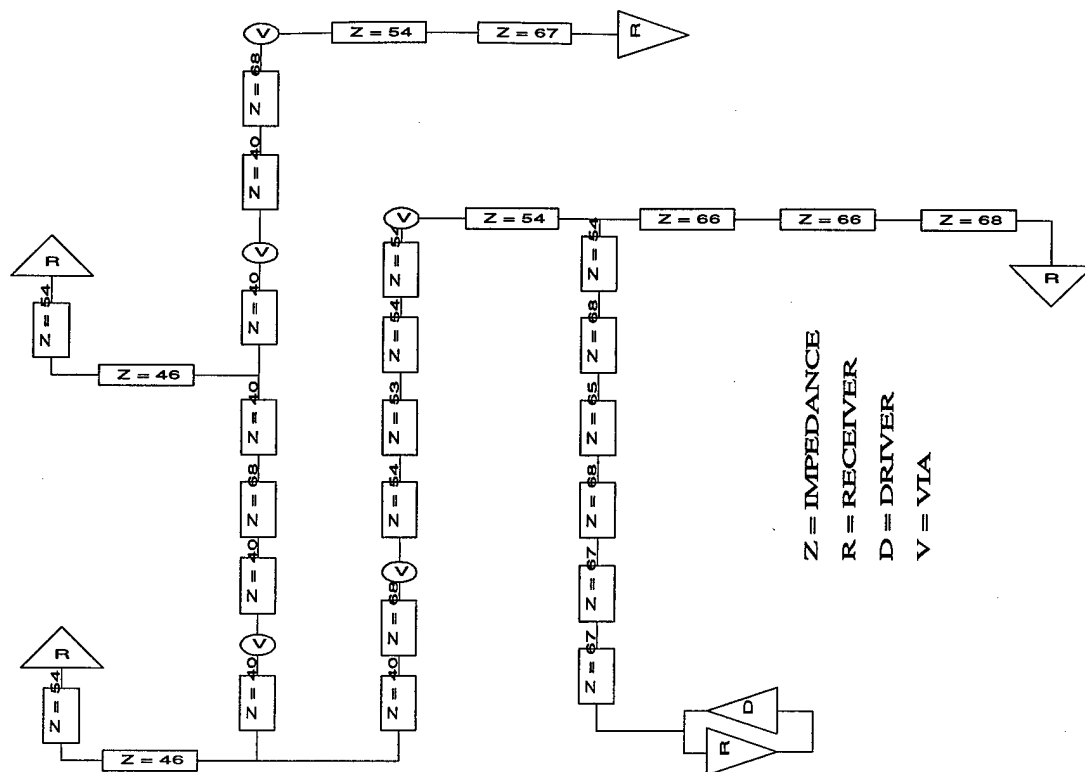


Figure 4. A typical multi-drop net model.

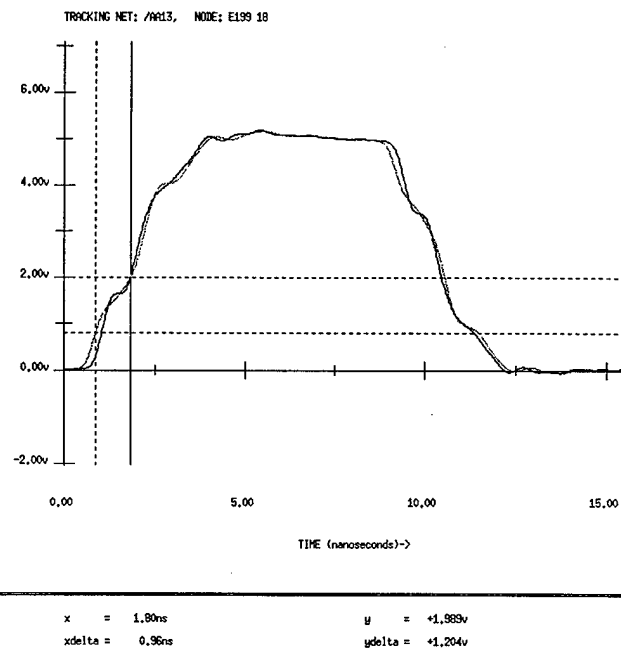


Figure 5. Electrical simulation of a net.

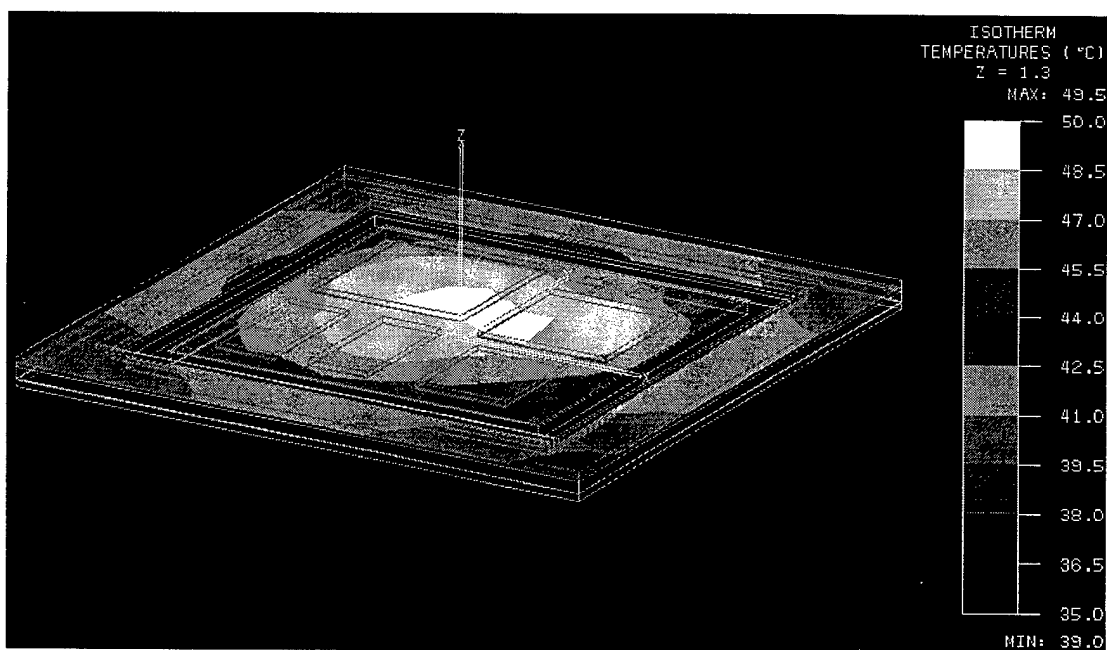
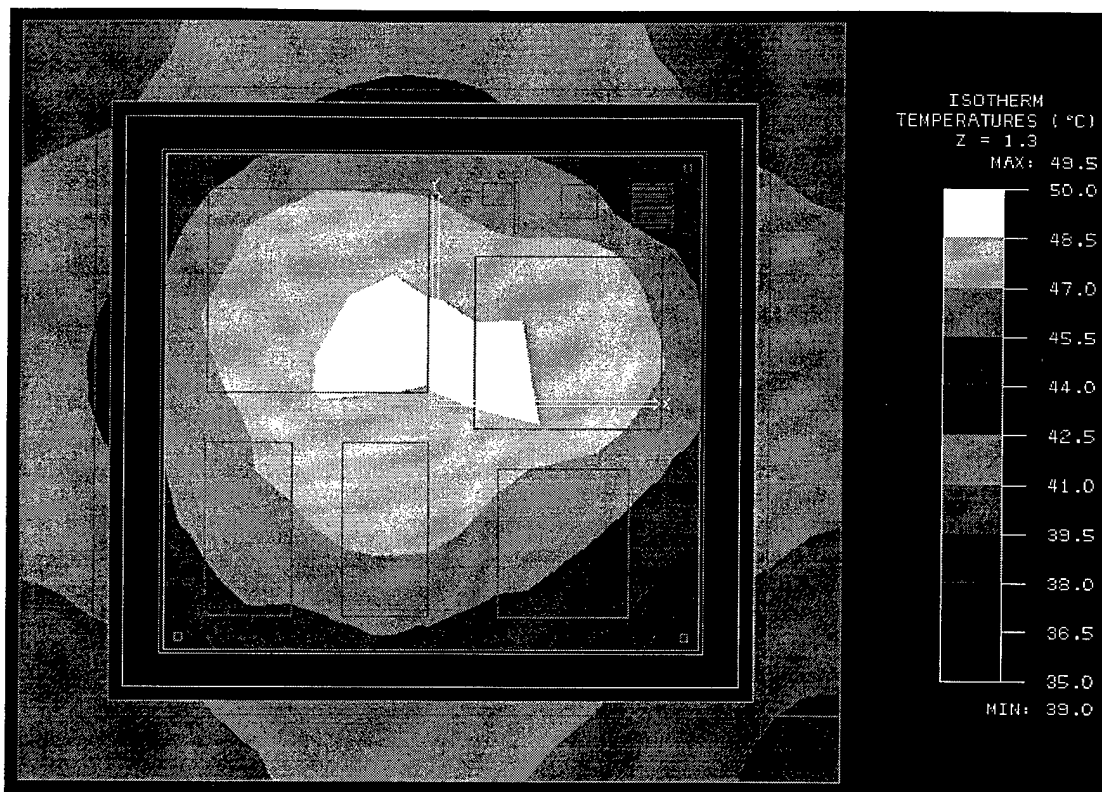


Figure 6. Thermal Analysis Simulation Results.

Dehkordi, P., Ramamurthi, K., Bouldin, D., and H. Davidson,
“Early Cost/Performance Cache Analysis of a Split MCM-Based MicroSparc CPU”,
Proceedings of 1996 IEEE Multi-Chip Module Conference, Santa Cruz, CA, February 5-7, 1996.

Early Cost/Performance Cache Analysis of a Split MCM-Based MicroSparc CPU

Peyman Dehkordi, Karthi Ramamurthi, Donald Bouldin¹
and Howard Davidson²

Abstract

Optimization of a microelectronic system is a difficult task involving a number of different disciplines. Often, an optimization in one discipline will result in a sub-optimal solution in other areas and the overall system. This paper looks into the optimization of a microelectronics system by concurrent consideration of the micro-architecture, package, and logic partitioning. This approach will attempt to identify an optimized design by helping the designer to explore the multi-dimensional solution space and evaluate the design candidates based on their system-level cost/performance. As a demonstration vehicle, we have evaluated the SUN MicroSparc CPU for possible MCM packaging based on sets of smaller dies using this approach. Cost/performance figure-of-merits are presented for various cache sizes using cost-optimized partitioning for flip-chip MCM-D packaging..

Introduction

Sub-micron IC technologies have low-latency device characteristics promoting higher clock speeds than in the past. Hence, interconnection delays are now becoming as important as the device delays. Microelectronic system designers have been taking advantage of these higher clock speeds to achieve higher performing systems by integrating as much functionality as possible into a single IC to reduce the interconnection penalties. However, the manufacturing yields of the resulting larger dies have been imposing constraints on the integration level. These constraints force the designers to perform various cost/performance trade-offs on the possible design candidates.

Advanced microelectronic packaging technology such as the multichip module (MCM) offers higher wiring density and shorter interconnect delays than the more traditional technologies such as printed circuit boards (PCBs). Conventionally, package technology is utilized by the package designer towards the end of the design cycle. Our previous work has shown that the packaging-related issues need to be considered throughout the design cycle by the system, IC, and package designers [1]. As we start to consider the packaging issues at an early stage of the design, we need to understand and evaluate the impact of

various packaging choices. The appropriate evaluation should be made at the system level by considering the packaging effects at various stages of the design cycle. As a designer makes this design paradigm shift, he will be faced with a different set of issues and constraints. This paper addresses these issues and outlines the approach that we have adopted in an attempt to solve this problem.

Case Study

Today's high performance microprocessors are highly integrated to achieve the best cost/performance. The designers are frequently faced with selection and trade-offs of various architectural issues [2]. As MCMs are becoming more available, the design architects are tempted to split a highly integrated "large" monolithic die into a set of smaller dies to achieve a better cost/performance [3][4]. This requires the designer to include cost/performance trade-offs between the IC and the MCM in the traditional analysis. As a demonstration vehicle, we have been evaluating and re-designing an MCM version of the SUN MicroSparc CPU based on a set of smaller dies. The monolithic baseline design is a 40 MHz RISC CPU with 2K of data cache and 4K of instruction cache as shown in Figure 1. The design consists of 750K transistors and the die measures 15 mm by 15 mm.

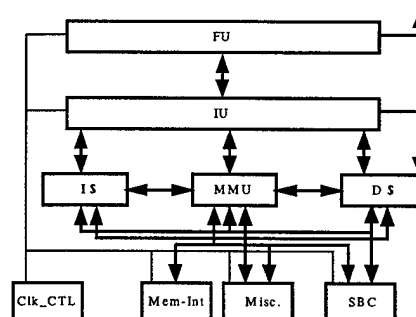


Figure 1. The MicroSparc CPU Block Diagram.

In this paper, we have extended our previous study by considering various sizes of first-level cache. We have evaluated a cost/performance figure of merit to assess the

¹ University of Tennessee, Knoxville, TN.

² Sun Microsystems, Mountain View, CA.

This work was sponsored in part by ARPA under grant DAAH04-94-G0004

best design candidates. Our efforts have focused on answering the following issues:

- 1) What is the near-optimum partition size (given a design and a packaging choice)?
- 2) What is the content of each partition?
- 3) What cache size will result in best cost/performance?
- 4) Should the ICs be fabricated using a combined logic/memory process or separate logic and memory processes?
- 5) How should all of the above issues be considered concurrently for a near-optimum design?

We have used *early-analysis* methods to evaluate various design choices by predicting their cost/performance as shown in Figure 2. Previous works such as [5] have looked at cache-analysis for MCM-based CPUs at a conceptual stage (pre-netlist phase of the design) using estimation-based models. Other work such as [6] evaluated the impact of dielectric and bonding technologies on GaAs MCM-based CPU for a fixed partitioning. Our approach also uses estimation-based models of [7] to predict the die and the package cost and performance in conjunction with a cost-optimized partition. This approach permits us to evaluate various designs quickly and to explore the solution space for a globally optimized design. We are performing detailed designs to validate the estimations models.

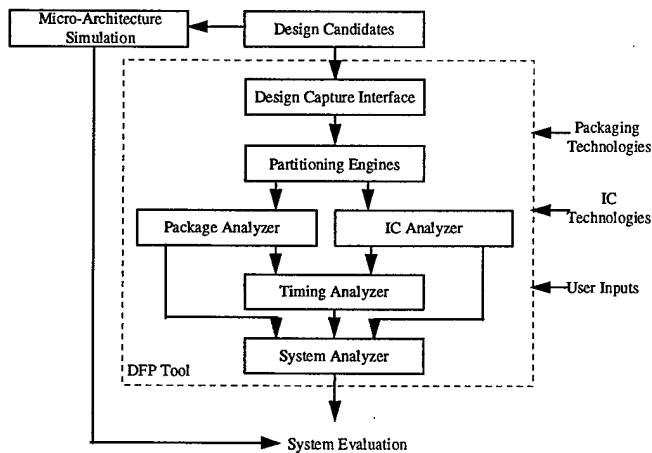


Figure 2. The Adopted Design Approach .

MCM Crossing Penalties

System performance is measured in terms of MIPS (Millions of Instructions per Second) and is defined as:

$$MIPS = \frac{1000}{T_c \times CPI}$$

where T_c is the cycle time (nS) and CPI is clock cycles per instruction. T_c can be impacted if the critical path is partitioned across multiple dies (as compared to the monolithic case). Figure 3a and 3b show the penalty for breaking a net and crossing the MCM substrate. The

interconnections on the MCM-Ds are characterized with wider traces, less resistive and lower dielectric constant as compared to the IC technologies. It has been argued that delays on the MCM interconnections are comparable to the IC and even better for longer nets [8]. The delay model is shown in Figure 3c and is similar to the ones in [9]. In order to manage properly the testing complexity for the final MCM, we have assumed all the dies have JTAG capabilities. The JTAG I/O cells add at least one 2-1 mux on each I/O pins which adds to the MCM crossing delay. Therefore the *minimum* delay for a partitioned-net across the MCM is:

$$T_{\text{net-breaking}} = 2 * T_{\text{jtag}} + T_{\text{buffer}} + T_{\text{mcm}}$$

$$T_{\text{net-breaking}} = 2 * T_{\text{jtag}} + T_{\text{buffer}} + T_{\text{mcm}}(D_{\text{min}}) + T_{\text{mcm}}(D_1 + D_2)$$

Where D_{min} is the minimum assembly spacing between the dies and D_1 and D_2 are distance between the I/O pad and the edge of the die. The first three terms are fixed overhead (for a given MCM and IC technologies) associated with the breaking of *any* signal net and the last term is a function of the die size, placement, and the MCM technology. The $T_{\text{net-breaking}}$ delay penalty can be reduced by optimizing the I/O buffers to reduce the T_{buffer} [10]. The timing simulations in this paper are based on the I/O buffer similar to one used in [11].

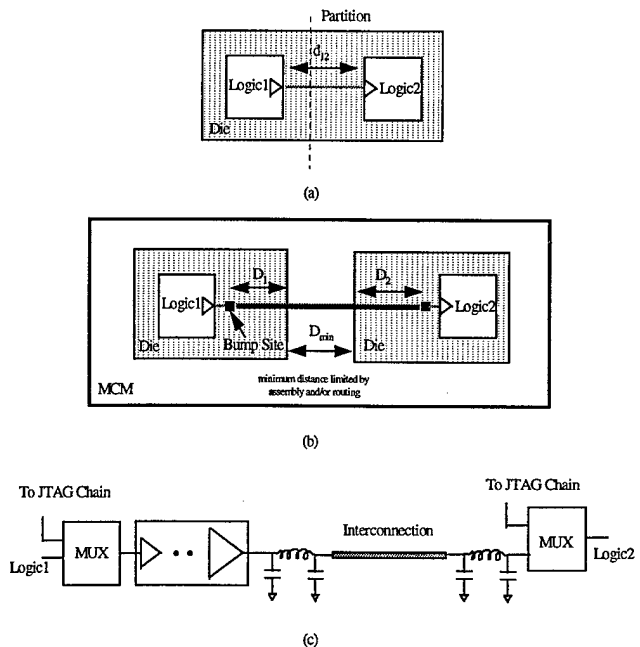


Figure 3. MCM Crossing Penalty.

Results

We have evaluated 2K/4K, 4K/8K, 8K/16K, and 16K/32K data/instruction cache sizes for the MicroSparc. We evaluated both the "combined logic/ memory" (LM) as

well as "separate logic/ memory processes" (L/M). The cost of the wafer and access time is assumed to be the same for both processes and the memory optimized process has double the density for SRAM implementation. There were a total of 8 different design candidates which were partitioned onto a MCM-D using flip-chip bonding. The partitioning stage was instructed to result in the lowest-cost set of dies for both the LM and L/M processes. For the LM process, the partitioner was allowed to mix or separate the logic and memory in order to reduce the cost (while using the LM cost model and density). For the L/M process, the partitioner was instructed to keep the logic and memory separate from each other (using both logic and memory cost models and densities). The partitioned results and the partition contents are shown in Table 1.

The results for the LM process show that the partitioner tends to separate the cache memory from the logic as the cache size becomes larger (to reduce the complete system cost: die + substrate + testing). Interestingly, this is also true for the 8K/16K and 16K/32K partitions based on the LM process in which the LM cost models were used (cache units are separated from logic). We decided not to consider 16K/32K LM because the partition result is the same as the L/M and one can always use the separate memory process for the stand-alone cache units.

The performance is measured in terms of the processor speed and CPI (cycle per instruction). The CPI measurements were provided by SUN using their trace simulator. The processor speed was calculated based on the delay which would occur if the critical path were broken. The MicroSparc CPU has a five-stage pipeline in which the critical path is between the cache and the TLB (translation lookaside buffer). We decided to take a pessimistic approach by considering every stage of the pipeline to be the critical path. We have also assumed the delay of the monolithic case is *all* due to logic (a pessimistic assumption but this is done for sake of simplicity and to offset any noise-related delays for the MCM cases). Therefore, a break in any of the pipeline stages (due to partitioning) will result in a longer machine cycle. The delay calculations are based on the estimated die-set size and their anticipated placement on the substrate. The die-set placements for all the cases are shown in Figure 3 along with the critical path representations.

The relative performance is shown in Figure 4 where all the figures of performance are normalized to the performance of the monolithic MicroSparc. Some important observations can be made from this analysis. The maximum clock frequencies in all the cases are lower than the monolithic CPU due to the substrate crossing penalty. Spice simulation shows that the worst-case delay is about 1 nS for the longest critical path on the substrate. For the 2K/4K MCM, both LM and L/M MCM-CPU's have 4% less performance (i.e. MIPS) than the monolithic design due to the substrate/buffer penalty. For all other cases, the MCM versions are higher performing by a factor of 3%, 7% and 12% for the 4k/8K, 8K/16K, and 16K/32K designs. The performance results also point out that there is not much

difference between MCMs using combined and separate processes (although they resulted in different partitions).

The relative cost of the MCM CPU's are shown in Figure 5 where all the costs are normalized to the cost of the monolithic MicroSparc. All MCM-CPU's have a much lower cost than the monolithic CPU (even the one with 16K/32K cache is 24% the cost of the monolithic CPU). Interestingly, the LM MCM-CPU's show a lower cost as compared to the L/M MCM-CPU's for 2K/4K and 4K/8K cache sizes. The L/M 2K/4K and 4K/8K MCM CPU's have the same cost. This is because of the high I/O connection to the cache units as compared to its small die size. Therefore, the die size was dominated by the I/Os rather than the cache size (we used a 250 micron pitch for the flip-chip bonding). This suggests that a more aggressive flip-chip technology (i.e. less than 250 micron) should be investigated since it *may* result in a lower cost system. We have also assumed that the I/O pads can be placed throughout the area of the memory dies. This may also not be a good practice because of the soft error due to C^4 α emission and regular SRAM layout. Therefore, the I/O pads may have to be placed on the perimeter of the die causing the size of the dies to be determined by the number of I/O rather than the memory cells for smaller cache memories.

Figure 6 shows the cost/performance figure-of-merit (i.e. cost per MIPS) where all the figures are normalized to the monolithic MicroSparc. The LM process shows a monotonic trend as the cache size increases. This suggests that the cost/performance gets worse for a combined logic/memory process as you increase the cache size. The L/M process reveals a curve where the minimum (i.e. lowest cost per MIPS) is at 8K/16K cache size. Interestingly the best and worst figure-of-merit is for 2K/4K and 8K/16K combined logic/memory CPU's respectively. Based on a given application requirement, one may select one of these MCM CPU's. For example, for a cost-sensitive application, the 2K/4K combined logic/memory process will be the choice. For highest performance, the 16K/32K based on separate logic/memory will be selected. For moderate cost/performance, the 8K/16K using separate logic/memory will be the choice.

Conclusions

As microelectronics packaging has been becoming the bottleneck of most high-performance systems, usage of advanced packaging such as MCMs is becoming inevitable. The packaging choices and their impact need to be considered and *evaluated* at an early stage of the design for the best system cost/performance.

The results of this work suggest that an MCM-based CPU may result in a *comparable* performance and *better* cost as compared to the monolithic CPU. Physical design of the substrate and optimized I/O drivers are the key to achieve cycle times comparable to the monolithic implementation. Signal integrity issues such as cross talk and ground bounce have to be evaluated carefully since partitioning results in more I/Os and interconnections on the substrate as shown in Figure 7.

Testing continues to be an important issue with MCMs in general. In this paper, the testing cost was modeled as a function of the I/Os. A more appropriate model will include the test coverage as well as the number of required test vectors. MCM-based split CPUs with JTAG dies may result in a better "controllability" and "observability" making testing cost *comparable* to the monolithic ones.

A separate memory process seems to result in a lower cost for *larger* cache sizes for this design. An optimized memory process results in a smaller die for a given memory size than the combined logic/memory process. As the feature size is scaled down, a small cache size will have a smaller footprint in a memory optimized process. In that case, the size of the memory die may be driven by the number of I/Os instead of the memory size.

Acknowledgments

The MSDA (Multichip System Design Advisor) tool has been extensively used throughout this work. The authors gratefully acknowledge the contribution of the MSDA tool which was originally developed by Peter Sandborn at MCC and is now being commercialized through Savantage, Inc.

References

- [1] Dehkordi, P., Ramamurthi, K., Bouldin, D., Davidson, H. and P. Sandborn, "Impact of Packaging Technology on System Partitioning: A Case Study", *Proceedings of the MCM Conference*, March, 1995.
- [2] Hennessy, J. L. and D. Patterson, *Computer Architecture, A Quantitative Approach*, Morgan Kaufmann Publisher, Inc., 1990.
- [3] Mendelsohn, A., "Will Monolithic or Multichip processor Win The Performance Race?", *Computer Design*, May, 1991.
- [4] Sechler, R., "RISC System/6000 Central Processing Unit (CPU) MCM", *Proceedings of International Conference and Exhibition Multichip Modules*, Denver, CO, April, 1993.
- [5] Roberts, J.D. and W. Dai, "Early System Analysis of Cache Performance for RISC Systems: MCM Design Trade-offs", *Technical Report UCSC-CRL-92-02*, University of California, Santa Cruz, March, 1992.
- [6] Kayssi, A., Sakallah, K., Brown, R., Lomax, R., Mudge, T., and T. Huff, "Impact of MCM's on System Performance", *Proceedings of Multichip Module Workshop*, Santa Cruz, CA, March, 1991.
- [7] Sandborn, P. and H. Moreno, *Conceptual Design of Multichip Modules and Systems*, Kluwer Academic Publishers, 1994.
- [8] Franzon, P., Stanaski, A., Tekman, Y., and S. Banerjai, "System Design Optimization for MCM", *Proceedings of Multichip Module Conference*, Santa Cruz, CA, January, 1995.
- [9] Bakoglu, H.B., *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley Publishing Company, Inc., 1990.
- [10] Gabara, T., Fisher, W., Knauer, S., Frye, R., Tai, K., and M. Lau, "An I/O CMOS Buffer Set for Silicon Multi Chip modules (MCM)", *Proceedings of Multichip Module Conference*, Santa Cruz, CA, March, 1993.
- [11] Taylor, M., Helman, D. and W. Dai, "AMP, A Bus-based Packet Switch Design Using MCM", *Proceedings of Multichip Module Workshop*, Santa Cruz, CA, March, 1991.

Cache Size	Logic/memory combined process (LM)		Logic/memory separate process (L/M)	
	Contents	Perf. Cost Cost/Perf.	Contents	Perf. Cost Cost/Perf.
2K/4K D\$/I\$	1 die (monolithic): (D\$, I\$, MMU, FU, IU, MEM-INT, SBC, CLK, MISC)	1.000 MIPS 1.000 \$ 1.000 \$/MIPS	N/A	
2K/4K D\$/I\$	3 dies: D\$, I\$, MMU FU, IU MEM-INT, SBC, CLK, MISC	0.961 MIPS 0.144 \$ 0.150 \$/MIPS	3 dies: D\$, I\$ FU, IU, MMU MEM-INT, SBC, CLK, MISC	0.960 MIPS 0.204 \$ 0.212 \$/MIPS
4K/8K D\$/I\$	3 dies: D\$, I\$, MMU FU, IU MEM-INT, SBC, CLK, MISC	1.025 MIPS 0.178 \$ 0.173 \$/MIPS	3 dies: D\$, I\$ FU, IU, MMU MEM-INT, SBC, CLK, MISC	1.024 MIPS 0.204 \$ 0.199 \$/MIPS
8K/16K D\$/I\$	4 dies: D\$ I\$ MMU, FU, IU MEM-INT, SBC, CLK, MISC	1.069 MIPS 0.238 \$ 0.223 \$/MIPS	3 dies: D\$, I\$ FU, IU, MMU MEM-INT, SBC, CLK, MISC	1.071 MIPS 0.213 \$ 0.198 \$/MIPS
16K/32K D\$/I\$	4 dies: D\$ I\$ MMU, FU, IU MEM-INT, SBC, CLK, MISC	N/A	4 dies: D\$ I\$ FU, IU, MMU MEM-INT, SBC, CLK, MISC	1.115 MIPS 0.238 \$ 0.214 \$/MIPS

Table 1. The Results of The Partitioned CPUs (All Values Are Normalized to the Monolithic CPU).

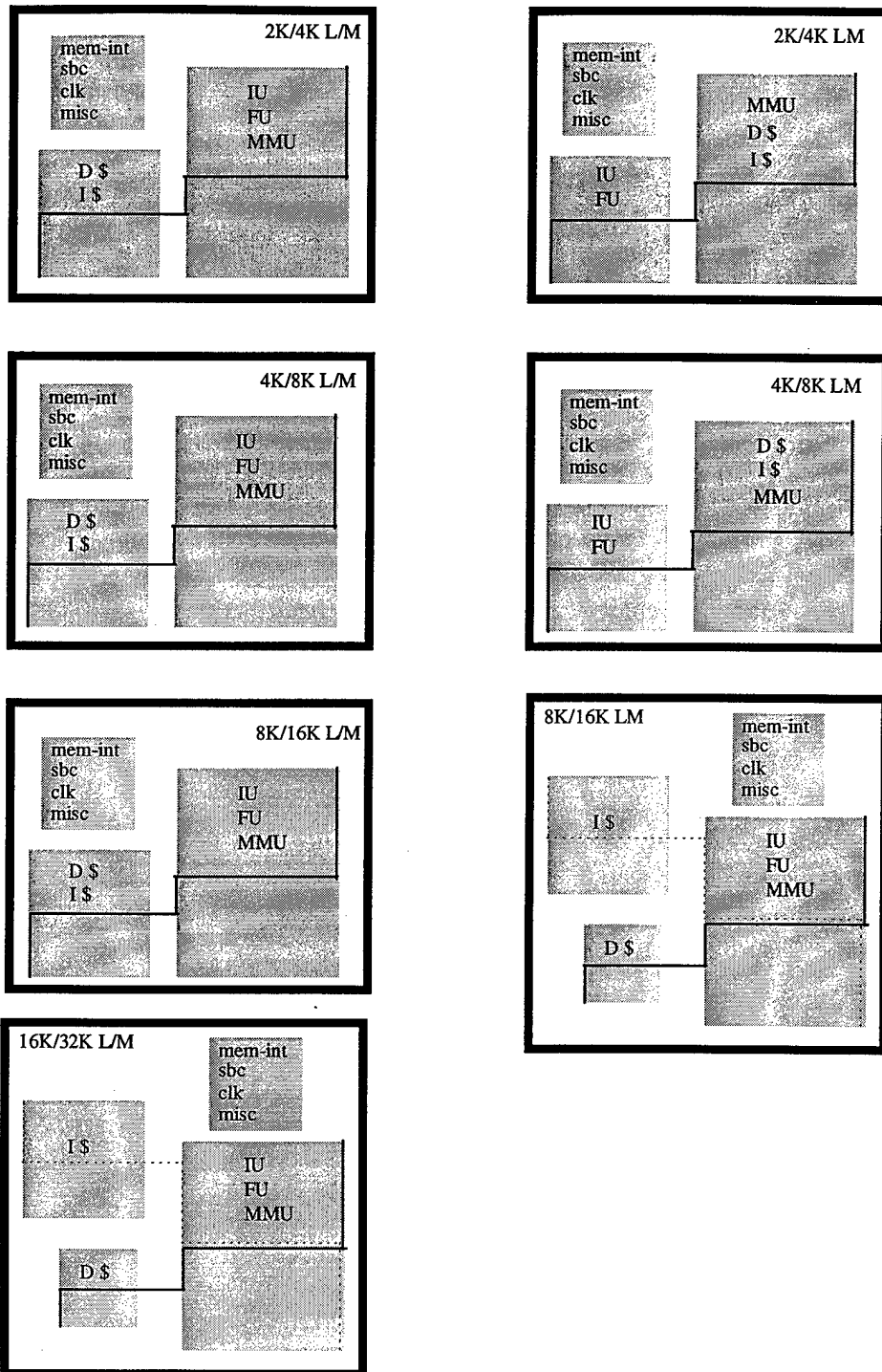


Figure 3. The Tentative Die-Set Placement on the MCM Substrate (Not to Scale).

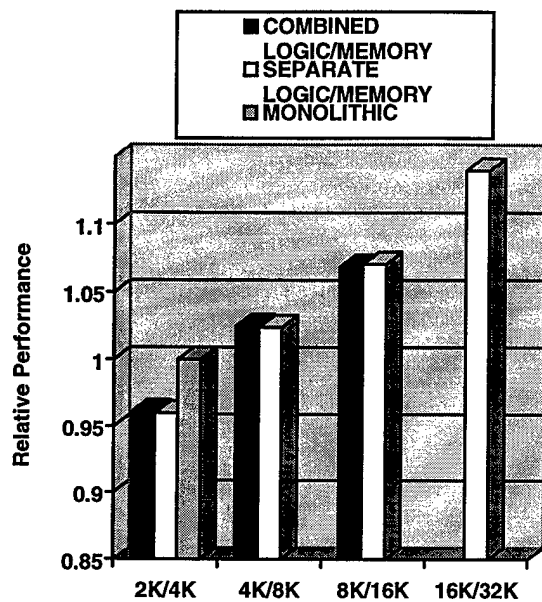


Figure 4. The Relative Performance Analysis.

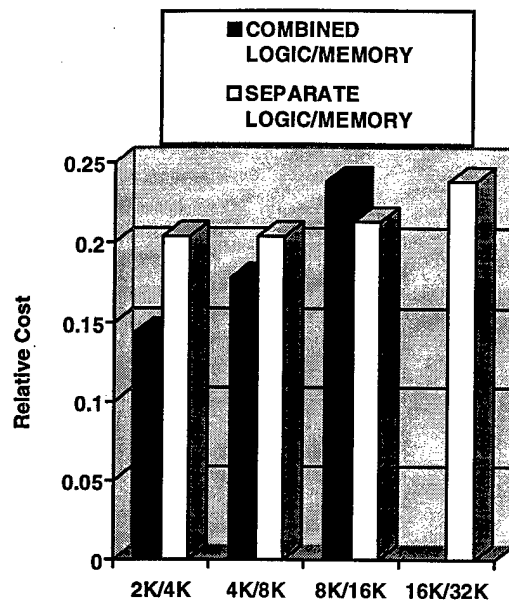


Figure 5. The Relative Cost Analysis.

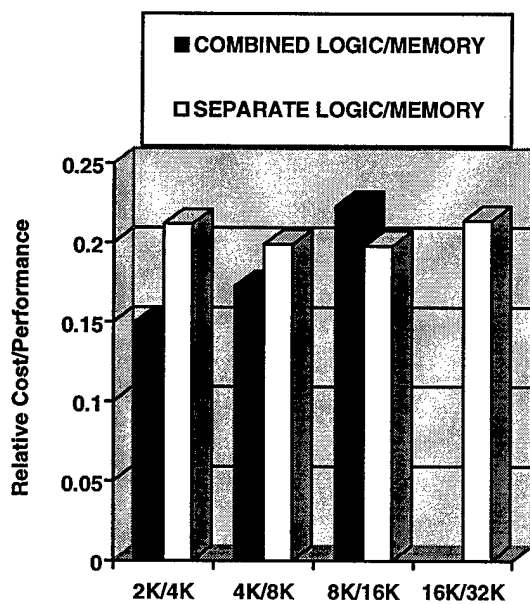


Figure 6. The Relative Cost/Performance Analysis.

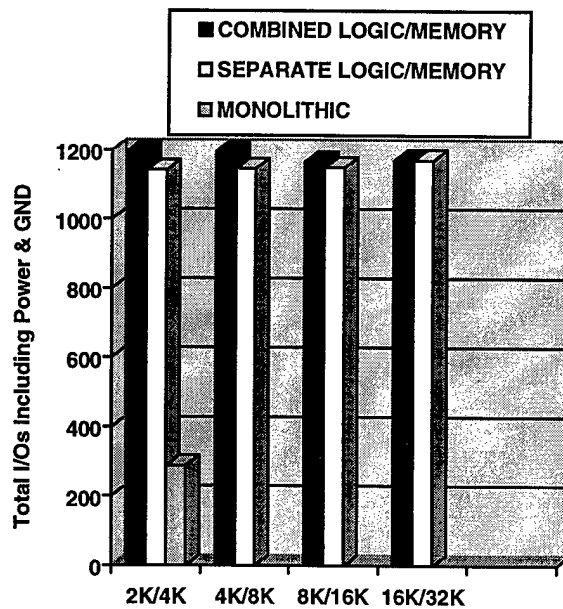


Figure 7. The Total Number of I/Os Including Power & Ground..

Dehkordi, P., Powell, T., and D. Bouldin,
“Performance Comparison of MCM-D and SMT Packaging Technologies for a DSP Subsystem”,
Proceedings of 1996 IEEE International Symposium on Circuits and Systems Conference,
pp. IV-245 thru IV-248, Atlanta, GA, May 12-15, 1996.

PERFORMANCE COMPARISON OF MCM-D AND SMT PACKAGING TECHNOLOGIES FOR A DSP SUBSYSTEM

Peyman Dehkordi, Tim Powell, and Don Bouldin

University of Tennessee

Knoxville, TN 37996-2100

dehkordi@microsys5.engr.utk.edu

tpowell@microsys6.engr.utk.edu

bouldin@microsys1.engr.utk.edu

ABSTRACT

Advanced packaging technologies such as MCMs offer superior performance as compared to the conventional PCB technologies. This paper discusses the design, development, and comparison of a general-purpose programmable DSP subsystem packaged in MCM and conventional surface-mount technologies. The subsystem contains a 32-bit floating-point programmable DSP processor along with 256 K-bytes of SRAM, 128 K-bytes of FLASH memory, a 10 K-gate FPGA, and a 6-channel 12-bit ADC. The complete subsystem has been interconnected on a 37 mm by 37 mm MCM-D substrate and is packaged in a 320-pin ceramic quad flat pack. This paper evaluates electrical and thermal performance for the MCM-D substrate and compares the results with the SMT version of the design.

1. INTRODUCTION

Today's sub-micron IC technologies have low-latency device characteristics that promote higher clock speeds than in the past. In high-performance systems, 50 percent of the total system delay is usually due to packaging and interconnect [1].

Advanced packaging technologies such as MCMs are emerging as a viable solution to this problem. Multichip modules provide housing and interconnect for multiple bare dies on a single high-density multi-layer substrate. Multichip module packaging has been in use by IBM and others for many years, but only recently has a low-volume and low-cost service become available to the design community [2].

As MCMs gain popularity, designers need to evaluate its cost/performance with the conventional technologies at the final system level to justify the insertion of such a technology. This paper evaluates the performance of a DSP sub-system based on MCM-D and surface-mount (SMT) technologies. The results (electrical and thermal) as well as technology descriptions are included in the

following sections. The paper concludes with a discussion on MCM related issues and conclusion.

2. FUNCTIONAL DESCRIPTION

The design is a DSP subsystem based on the Motorola 96002 32-bit programmable floating-point DSP processor. In addition, the module includes 256 K-bytes of SRAM, a 128 K-byte EEPROM, a 10 K-gate Xilinx 4010 FPGA, and a 6-channel 12-bit ADC. The module was designed to exploit the multi-processing capability of the 96002. The FPGA is primarily a hardware pre-processor of incoming data but also provides other generic logic functions required for implementation of the entire subsystem.

3. PACKAGING TECHNOLOGIES

Two packaging technologies have been evaluated: MCM-D and SMT. The characteristic of both technologies are shown in Table 1 and Table 2 respectively. The MCM-D substrate is a 37 mm x 37 mm 5-layer (1 pad, 2 power, and 2 routing layers) substrate based on the *MicroModule Systems Merged Via D500* process [3]. Copper is used for the routing layers and positive power plane, and a 50 mil aluminum substrate serves as a ground plane. The top layer is the pad and die connect layer. The die are attached to the thin film interconnect using epoxy and are wirebonded to pads using gold wire. For the top layer of metallization, no routing is allowed, and all pads have vias that connect to the appropriate signal layers or power planes. The MCM is packaged in a cavity-down, 320-pin ceramic quad flat pack with preassigned power and ground pins.

The SMT board is 6.5 inches long and 3.75 inches wide with 6 metallization layers (2 power planes and 4 trace layers). Two of the trace layers are on the PCB top and bottom surfaces and share area with the components. The remaining two trace layers are internal and sandwiched between the power planes. All metallization layers are standard 1 oz. copper, and the dielectric regions of the board are constructed using standard 10 mil FR-4 glass epoxy. In order to facilitate the high interconnect

demand for this design, buried vias are used to connect between layers. PCB is assumed to have characteristics as described by [4].

4. ELECTRICAL ANALYSIS

Figures 1 and 2 show the final routed MCM and SMT layouts. Table 3 lists the area efficiency statistics of the MCM and SMT implementations. A complete detailed comprehensive analysis is presented in [5].

The entire design was simulated with *QUAD's XNS* tool [6] to identify the limitation of both technologies. Fast I/O driver models (rise and fall times of approximately 1 to 4 nsec.) were used to simulate all digital networks and I/O for both design representations. Several simulations were performed using drivers with unloaded rise and fall times ranging from 4 nsec to 1 nsec. Table 4 lists the worst case results for the simulations performed using 1 nsec rise/fall time drivers (limitation of the design without termination resistance).

The MCM-D substrate was found to be much faster than the SMT PCB. The decrease in net delay for the MCM-D can be attributed to several factors including the removal of single-package pin parasitics and reduced parasitic net capacitance. For PCBs, single-package pins contribute inductance and capacitance which can seriously distort and delay signals. When the bare die are placed on the MCM-D substrate, the single-package pin parasitics are replaced with wirebonds which have much less parasitic capacitance and inductance. In addition, the MCM-D interconnect traces are much smaller and shorter, and therefore, they contribute much less parasitic net capacitance.

The MCM-D was also found to have less crosstalk between traces than the PCB. The crosstalk was only calculated on the plane of the routed trace. Other routing planes which could cause contributions from intersecting traces were not considered. The reduced crosstalk for the MCM-D can be attributed to shorter segments of parallel traces and a reduced thickness in trace metal compared to the SMT PCB. Crosstalk is a result of capacitive coupling and inductive coupling between traces. By reducing the length of parallel trace segments, the mutual capacitance and inductance are reduced. In addition, the MCM-D traces are 4 μm thick compared to the PCB trace thickness of 1.4 mil (35.6 μm). So, the effective area of potential coupling is also reduced.

Next, overshoot and undershoot measurements are considered. Again, for 1 nsec drivers, the MCM-D substrate resulted in improved signal performance. Overshoot and undershoot are directly related to driver rise and fall times and the characteristics of the network and receivers being driven. For the MCM-D version, the single-package pin parasitics have been removed, and the

traces are less capacitive. The result is a cleaner and better behaved signal as indicated in Figure 3. The results listed in Table 4 indicate that the SMT PCB would be border line to inoperable when using 1 nsec drivers because of overshoot and undershoot problems. Although the MCM-D values are slightly high, they are still within the acceptable operating range of all devices in the design.

Power distributions in both systems are compared in terms of supply bounce. Supply bounce is related to the inductance of the power distribution network. Supply bounce results when a large amount of current is passed through an inductive network in a very short time. Table 4 shows the simulation results for both MCM-D and SMT. These supply bounce simulations were performed with all possible drivers switching simultaneously and no decoupling capacitance was considered. The results which are Pessimistic (since no decoupling capacitances are included in the simulation) indicate that the MCM-D substrate will have less supply bounce than the SMT PCB substrate in worst case condition (driver with 1 nsec. rise time). For the PCB, the supply bounce is higher because of the parasitic inductance of the single-package pins. In single-package components, power/gnd pins on the package are inductive. In addition, multiple power and ground pads are often bonded to a single pin resulting in a high inductive path for current during signal transition. When bare die are placed on a MCM substrate, each pad can be bonded separately significantly reducing inductance resulting in lower supply bounce. Also, supply bounce can be reduced with the addition of decoupling capacitance. Decoupling capacitors can supply charge to drivers during simultaneous transitions when the inductance of the power network prevents large instantaneous current influx. For PCBs, it is considered routine procedure to include a significant amount of decoupling capacitors in cases where supply bounce could result. For the PCB design in this paper, decoupling capacitors occupy the bottom side of the SMT board. For the MCM-D design, there is an inherent low inductive decoupling capacitor (approximately 10 nF) present between the power and ground plane because of the construction of the MCM substrate.

5. THERMAL ANALYSIS

MCMs integrate several large VLSI dies consuming several Watts within a small area. Thermal analysis of an MCM is very important to ensure a reliable operation. The objective of thermal analysis is to predict the junction temperature of the dies in determining long-term reliability of the components and the module. The results for the MCM and PCB were generated using three-dimensional finite-element mesh analysis at 25 °C in still

air. For the MCM-D, the thermal path from the die to heat sink is comprised of the following three paths:

Path1 = $R_{die} + R_{die\ epoxy} + R_{interconnect}$

Path2 = $R_{MCM\ package} + R_{interconnect\ epoxy}$

Path3 = $R_{thermal\ grease} + R_{heat\ sink}$

Thermal vias were placed under all the dies to improve the thermal conductivity path from the dies to the backing plate. Thermal vias were created in such a fashion to allow two traces between them. This resulted in less routing penalty under the dies due to the thermal vias. The thermal parameters and results for the MCM-D are listed in Table 5. The maximum junction temperature was simulated to be 50 °C at the FPGA which is well within the maximum rating of the dies.

The thermal parameters and results for the PCB are listed in Table 6. The simulations assumed no heat sink on any of the components. The maximum junction temperature was simulated to be 62 °C at the FPGA which is well within the maximum rating of the dies.

6. MCM-RELATED ISSUES

With the evolution of MCM technology, it is permissible for engineers to consider a design flow incorporating MCM technologies. This allows the designer to make better decisions concerning the physical implementation of the design. Early-analysis tools such as [7] offer designers quick comparisons of different packaging technologies for their specific application.

However, there are still problems in obtaining Known Good Die. These issues include potentially long lead times and small selection of bare die for a given design. More effort needs to be expended in the release of I/O driver simulation models (IBIS or SPICE) by the IC vendor to ensure proper simulation results.

With MCMs, testing can be a problem since the lines and pins are not easily probed. Therefore, design for testability (DFT) techniques must be incorporated from an early phase of the design. Using boundary-scan components can help with the testing; however, not all the components have boundary-scan capabilities.

7. CONCLUSION

Multichip modules are becoming a viable choice of packaging for high performance/miniaturized electronics. This technology is no longer considered to be an exotic technology and is being used in telecommunications, consumer electronics and workstations. MCMs offer superior performance as compared to the conventional technologies (packaged-part on PCB). An appropriate system-level cost/performance comparison (along with risk factor) of MCM with the conventional technologies

may justify the usage and insertion of such a technology. Availability of the bare dies (and their I/O buffer models) and testing issues will be challenging issues to be considered.

REFERENCES

- [1] Bakoglu, H.B., *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley Publishing Company, Inc., 1990.
- [2] Dehkordi, P., Powell, T., and D. Bouldin, "Development of a DSP/MCM Subsystem: Assessing Low-volume, Low-cost MCM Prototyping for Universities", *Proceedings of Multichip Module Conference*, February, 1996.
- [3] MicroModule Systems Corp., *MCM-D Technology Kit, Rev 1.01*, MicroModule Systems Corp., Cupertino, CA, 1994.
- [4] Prasad, R., *Surface Mount Technology: Principles and Practice*, Van Nostrand Reinhold, New York, 1989.
- [5] Powell, T., "The Development, Design, and Comparison of Printed Circuit Board and MultiChip Module Representations of a DSP Subsystem", *Master's Thesis*, University of Tennessee, May, 1996.
- [6] Mentor Graphics Corporation, "Transmission Line and Crosstalk Products User's Manual XNS-TLC-XFX Software Version 8.4_1", Mentor Graphics Corporation, Wilsonville, OR, 1994.
- [7] Sandborn, P. and H. Moreno, *Conceptual Design of Multichip Modules and Systems*, Kluwer Academic Publisher, 1994..

MCM-D SUBSTRATE CHARACTERISTICS	
Substrate Property	Value
Dielectric	Polimide
Dielectric Constant (Er)	3.5
Metallization	Copper
Copper Resistivity	1.7E-6 Ohms-cm
Number of Trace Layers	2
Number of Power Planes	2 (VCC and Ground)
Board Thickness	1.3 mm (52 mils)
Component Mounting	Wirebonded (Top Surface Only)
MCM-D TRACE CHARACTERISTICS	
Signal 1 Trace Property	Value
Width	20 um (42 um spacing; 62 um pitch)
Characteristic Impedance	58 Ohms
Capacitance	1.0 pF/cm
Inductance	3.4 nH/cm
Resistance	2.0 Ohms/cm
Propagation Delay	58 ps/cm
Signal 2 Trace Property	Value
Width	16 um (46 um spacing; 62 um pitch)
Characteristic Impedance	50 Ohms
Capacitance	1.2 pF/cm
Inductance	3.0 nH/cm
Resistance	2.4 Ohms/cm
Propagation Delay	60 ps/cm

Table 1. MCM-D Properties

SMT BOARD CHARACTERISTICS	
Board Property	Value
Dielectric	10 mil FR-4 Glass-Epoxy
Dielectric Constant (Er)	4.5
Metallization	1.4 mil Copper
Copper Resistivity	1.7E-10 Ohms-cm
Number of Trace Layers	4
Number of Power Planes	2 (VCC and Ground)
Board Thickness	60 mils
Component Mounting	Both Sides
SMT TRACE CHARACTERISTICS	
Surface Trace Property	Value
Width	8 mils (8 mil spacing; 16 mil pitch)
Characteristic Impedance	73 Ohms
Capacitance	0.767 pF/cm
Inductance	4.1 nH/cm
Resistance	23.5 mOhms/cm
Propagation Delay	56 ps/cm
Buried Trace Property	Value
Width	8 mils (8 mil spacing; 16 mil pitch)
Characteristic Impedance	55 Ohms
Capacitance	1.3 pF/cm
Inductance	3.9 nH/cm
Resistance	23.5 mOhms/cm
Propagation Delay	71 ps/cm

Table 2. PCB Properties

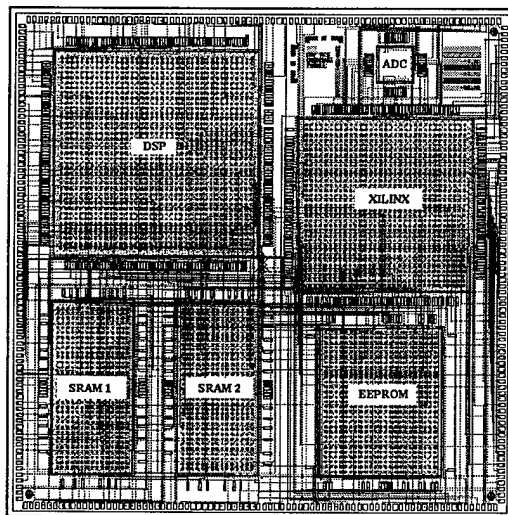


Figure 1. MCM-D Layout

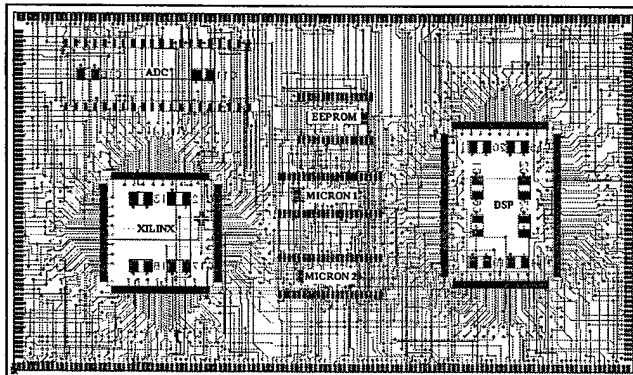


Figure 2. SMT PCB Layout

MCM-D COMPONENT DIMENSIONS				PCB COMPONENT DIMENSIONS			
Component	Width (mm)	Length (mm)	Surface Area (mm ²)	Component	Package Width (mm)	Package Length (mm)	Package Surface Area (mm ²)
DSP 96002	15	15	225	DSP 96002	28	40	1120
Xilinx 4010	12.88	12.61	162.42	Xilinx 4010	28	28	784.00
SRAM (2)	5.842	12.725	74.34	SRAM (2)	10.2	28.6	291.72
EEPROM	9.042	10.897	98.53	EEPROM	12.2	21	256.20
ADC	2.4	2.4	5.76	ADC	15	51	765
MCM-D SUBSTRATE	37	37	1369	SMT Board	95.25	165.1	15725.78
Total Chip Area (mm ²)			640.39	Total Package Area (mm ²)			3508.64
Top Surface Area Efficiency (Chip Area/MCM Area)			46.78%	Chip Area/MCM Area			4.07%

Table 3. Area Results

SIMULATION RESULTS					
	Net Delay	Crosstalk	Overshoot	Undershoot	Supply Bounce
MCM-D	1.44 nsec	214 mV	0.32 V	-0.23 V	0.65 V
PCB	5.0 nsec	479 mV	0.62 V	-0.59 V	1.4 V

Table 4. Simulation Results

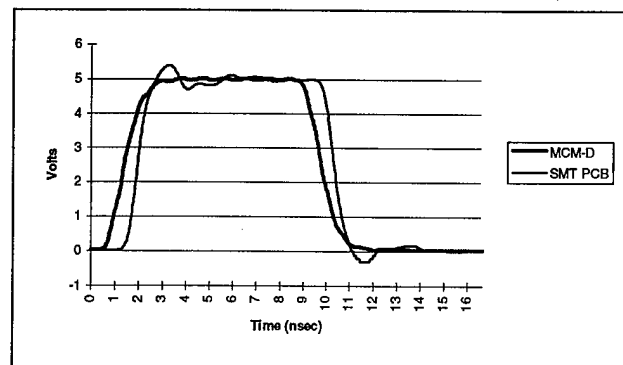


Figure 3. MCM and SMT PCB Signal Waveforms

MCM-D COMPONENT TEMPERATURES				
Component	Temperature Results (Celsius)			
	Minimum	Maximum	Average	Junction
DSP 96002	46.5	49.3	47.9	48
Xilinx 4010	47.2	49.5	48.3	48.3
SRAM (1)	45.5	47.7	46.6	46.6
SRAM (2)	46.6	48.5	47.4	47.4
EEPROM	44.5	47.1	45.8	45.8
ADC	46	46.2	46.1	46.1

Table 5. MCM-D Thermal Results

PCB COMPONENT TEMPERATURES				
Component	Temperature Results (Celsius)			
	Minimum	Maximum	Average	Junction
DSP 96002	56	59.9	58.9	59
Xilinx 4010	56.3	61.7	60.5	60.7
SRAM (1)	57.8	60.4	59.7	59.9
SRAM (2)	57.9	60.6	59.7	60
EEPROM	50.8	53.2	52.5	52.6
ADC	42.5	46.8	44.5	44.5

Table 6. PCB Thermal Results

Dehkordi, P., Chandra, T. and D. Bouldin,
“Intrinsic Area Array ICs: What, Why, and How”,
Proceedings of 1997 IEEE Multi-Chip Module Conference, Santa Cruz, CA, February 4-5, 1997.

Intrinsic Area Array ICs: What, Why, and How

Peyman Dehkordi, Chandra Tan, and Donald Bouldin *

Department of Electrical Engineering

The University of Tennessee

Knoxville, TN 37996-2100

{dehkordi,chandra,bouldin@microsys1.engr.utk.edu}

Abstract

Area-array bonding technology (i.e. Flip-chip, C4) was pioneered by IBM in the late 1960's as an alternative to periphery bonding technology (i.e. wire-bond). In recent years, several commercial companies have started offering bumping and flip-chip services. Flip-chip technology is expected to grow at a compound annual growth rate of 38% through the year 2001 [1]. The purpose of this paper is to address the IC design issues and alternatives that are presently being used for area-array bonding technology and show the impact of these design issues at the system level.

1. Introduction

Area-array solder interconnections can be made using the flip-chip assembly technique. Known also as the C4 process (Controlled Collapse Chip Connection), flip-chip bonding was developed by IBM in 1964 and has been used for over 30 years to assemble IBM's mainframe computer modules [16]. Flip-chip gives the highest chip density of any packaging method to support pad-limited IC designs. Instead of placing the chips in space-wasting individual packages, flip chip places them face down onto matching connections on a substrate or board by means of solder pads or bumps. Since the connections are under the chip, no additional space is required for bond wires. Connections may be made anywhere on the surface of the chip, using the whole chip area for connections, instead of just around the perimeter. The solder bumps provide shorter connections avoiding the performance penalties of conventional bond wires.

Some studies have been conducted recently to show the impact of area-array flip-chip bonding on today's VLSI designs. The study conducted in [7] showed the reduction of die size and the increase in I/O count when peripheral

wire-bond technology was replaced by area-array flip-chip technology. A conceptual trade-off analysis between peripheral and area-array bonding in MCMs is presented in [14]. The impact of partitioning an ultra-large single die into multiple smaller dies housed on a MCM is shown in [8]. The result of this study indicated a dramatic reduction of cost if the CPU studied were divided into three chips bonded using area-array technology and interconnected on a MCM-D substrate.

In recent years, several commercial companies have started offering bumping and flip-chip services. Flip-chip technology is expected to grow at a compound annual growth rate of 38% through the year 2001 [1]. Flip-chip technology does have a major drawback in that not every IC design is built using this process, so a decision to use flip chip for all the devices in multichip modules or boards may require postprocessing the die to put solder bumps on the area-array pads. This postprocessing step involves the placement and routing of the area-array pads on the existing top metal layer of the die or on an additional distribution metal layer. The purpose of this paper is to address the IC design issues and alternatives that are presently being used for area-array bonding technology and show the impact of these design issues at the system level.

2. Present Approaches for Designing An Area-Array IC

Recently, most military and commercial electronic products are evolving toward lower cost, smaller form factor, lower weight and higher performance. All the improvements can best be realized at the chip level by eliminating the IC packages and transitioning from perimeter to area I/O. With the exception of IBM and a few semiconductor manufacturers who are producing area-array ICs, most manufacturers are reluctant to add bumps to their dies or redesign for area-array unless large quantities of dies are procured.

*The authors gratefully acknowledge the support of ARPA grant DAAH04-94-G0004

Presently, there are two major approaches for designing an area-array IC: **Intrinsic** and **extrinsic**. An intrinsic area-array IC is one specially designed and laid out for area-array bonding. Whereas, an extrinsic area-array IC is one originally designed and laid out for periphery bonding but has been converted for area-array bonding by means of a redistribution layer. As shown in Figure 1, both types of ICs take advantage of the superior electrical performance of solder bumps and a smaller foot print as compared to wire-bond ICs. Our previous work quantitatively showed additional benefits can be gained by using intrinsic area-array ICs rather than using the extrinsic counterparts at the IC level [6].

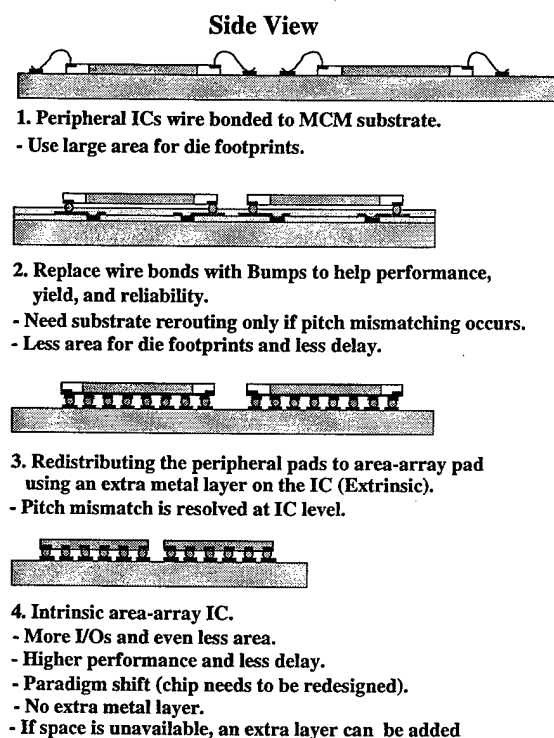


Figure 1. Why Intrinsic Area-Array IC.

An intrinsic approach, pioneered by IBM [3, 13, 15], is to place and route the I/O buffers anywhere closer to the center of the die and connect these I/O buffers to their corresponding pad sites placed in an array on the surface of the die. Using this approach, it is possible to design chips around circuit constraints rather than pad constraints; that is, it is possible to place buffer circuits where they make the most sense from a circuit design standpoint rather than where they are required for periphery bond purposes. Although the area-array IC resulting from this approach is fully optimized for silicon performance, this approach requires many additional considerations to conventional IC

physical design processes. The extrinsic approach is the semi-optimized approach which is a process for transforming fine pitch peripheral IC pads to area array pads using a redistribution layer. In this approach, any die with periphery pads can be converted to an area-array IC by employing this redistribution process. This is best done at the wafer stage where a low-cost, batch photolithographic process can be used. Several semiconductor manufacturers including Motorola have designed some of their chips with area-array flip-chip assembly using this process [10, 11]. Some algorithms have been developed to perform the routing from the peripheral bonding pads to the area array C4 pads [4, 5, 17].

3. Intrinsic Approach

3.1. Problem Definition

At the present time, there is no commercial IC layout tool available to the IC design community to perform intrinsic area-array pad placement and routing. We are currently finishing the development of an area-array pad router which automates the placement and routing of the area-array pads on the IC. This is a post-processing tool which allows IC layouts generated by any other tool to be processed for area-array I/O pad placement and routing. This tool is different from a previously reported tool [5] since ours will place and route the pad using existing layers on the IC (if possible) before adding a new redistribution layer. Figure 2 shows some of the design steps for the intrinsic area-array IC layout problem. Unlike the physical design of a periphery

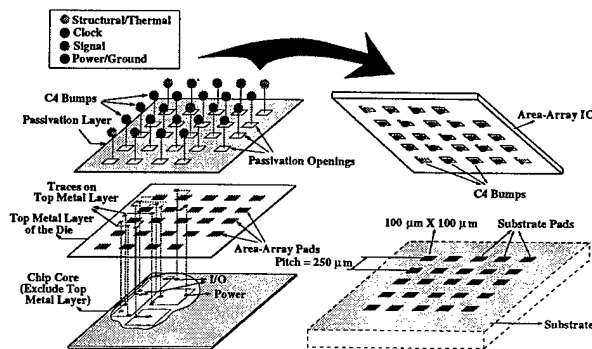


Figure 2. Intrinsic Area-Array IC Layout Problem Definition.

bonded IC which places and routes the I/O cells (pads and I/O buffers) around the die after the active components of the chip have been laid out, the I/O buffers for area-array pads and the active components are laid out at the same time.

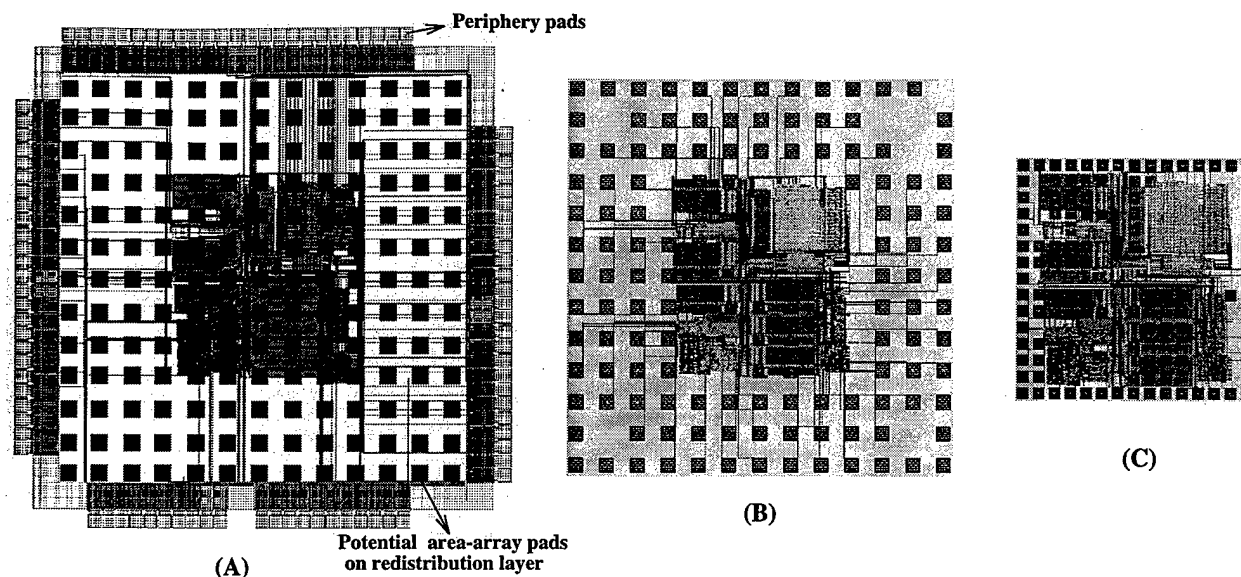


Figure 3. Extrinsic Vs Intrinsic Area-Array IC.

The area-array bonding pads will then be placed on the top metal layer of the die and routed to their corresponding I/O buffers.

The layout problem of an intrinsic area-array IC can be stated as: Given a "core" portion of the chip which already contains the I/O buffers, place the possible uniformly spaced area-array pads on the top metal layer of the design which may contain some prerouted wires and keepout areas and route these pads to the I/O ports of the chip using a limited number of available connection layers, such that design rules are obeyed and some other objective functions are satisfied.

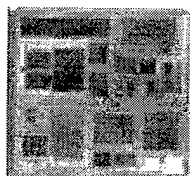
3.2. Implementation

The framework for intrinsic area-array IC placement and routing has been implemented in C and run on a SUN Sparcworkstation. Given the mask geometries of the chip core stored in CIF (Caltech Intermediate Form), our tool will generate the area-array padframe of the IC. The pads are large areas of metal that are left unprotected by the overglass layer so they can be soldered to the bumps using the C4 process. Pad size is defined usually by the minimum size to which the solder bumps can be attached. The spacing of the pads is defined by the minimum pitch of the substrate. Both of these parameters are specified by the designer when running our tool. Before the tool proceeds to pad placement and pad routing steps, the chip core layout is flattened to extract the geometric information on the top few layers (the number of layers is specified by the designer), the pad keepout areas, the locations of I/O terminals, and the boundary of the core

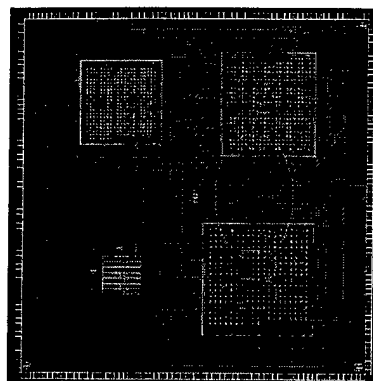
circuits. Utilizing this extracted information, the pad placement algorithm calculates the center of the chip core and then using this center as a reference point, generates a set of the potential area-array bonding pads on the top metal layer satisfying the user-specified bond pitch. The power/ground and clock I/O pads are routed first before the signal I/Os. The routing technique used is based on the general area routing algorithm [2, 9, 12].

4. Results and Discussions

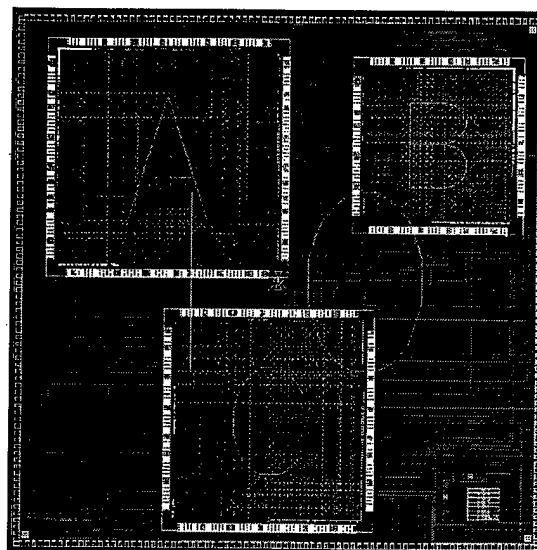
In this paper, we present the results our area-array layout tool produced for two examples to show the impact of the intrinsic area-array IC on the IC and system level. The first example is the "Quantizer" taken from the tutorial example of the Epoch physical design tool. Figure 3A shows the "Quantizer" die with peripheral bonding pads. The size of the die is $3.88 \times 3.87 \text{ mm}$. The number of peripheral I/O pads is 96 with size of $100 \mu\text{m}$ and pitch of $110 \mu\text{m}$. The core of this die is $1.48 \times 1.64 \text{ mm}$. Area-array pads have been superimposed on the top surface of the die to illustrate how the area-array version of the die could be done using the extrinsic approach to redistribute the I/Os from the periphery to an area-array. The size of each of these pads is $100 \mu\text{m}$ and the spacing between pads is $250 \mu\text{m}$. Figure 3B shows the area-array version of the die using the intrinsic approach with the same pad size and pitch used in extrinsic approach. The area pads were placed on the existing top metal layer and the pad routing was done on the top two metal layers. The die size is reduced ($2.85 \times 2.85 \text{ mm}$) and the number



(A) MONOLITHIC
15 mm



(B) AREA-ARRAY MCM
27 mm



(C) PERIPHERAL-MCM
47 mm

Figure 4. System Level Comparison (Monolithic, Intrinsic, Wire-bond.)

of I/O pads is increased (118). The extra I/O pads were used to connect more power and ground to the die to improve the signal integrity. With the advances of flip-chip technology, smaller bonding pad size and pitch are possible in the future. The design with area pad size of $80\ \mu\text{m}$ and pitch of $120\ \mu\text{m}$ is shown in Figure 3C. The reduction of the die size is more profound. If we use extrinsic approach with the same pad size and pitch, there will be no reduction of die size. In the above examples, we have shown the impact of intrinsic approach on the IC level.

In the next example, we will show the impact of the intrinsic area-array IC on the system level. We used the SUN MicroSparc CPU as a representative of a large design where MCM implementation may be required. The MCM implementation for a 3-chip design is shown in Figure 4 along with the monolithic solution (drawing to scale). The substrate is based on the Micro Module Systems MCM-D process and the ICs are Hewlett-Packard 0.5-micron technology as offered through the MIDAS and MOSIS services. Figures 4B and 4C are the 3-chip MCM system based on the intrinsic and periphery ICs. The results illustrate a clear advantage of the intrinsic approach. The study conducted in [8] indicated in more detail the impact of area-array technology at the system level.

5. Conclusion

We have presented an intrinsic approach of designing an area-array IC. We described the major reasons why we needed this approach and showed how this approach worked.

The preliminary results on small and large designs have shown the impact of this intrinsic approach on the IC and system level.

6. Acknowledgement

The Epoch Physical Design tool by Cascade Design Automation and the MCM Station software by Mentor Graphics have been extensively used throughout this work. The authors gratefully acknowledge the donation of these two tools to the University of Tennessee.

References

- [1] In *World Wide Web home page of FlipChip Technologies*, <http://www.flipchip.com>, August 1996.
- [2] M. H. Arnold and W. S. Scott. An Interactive Maze Router with Hints. In *Proc. 25th Design Automation Conf.*, 1988.
- [3] A. J. Blodgett and D. R. Barbour. Thermal Conduction Module: A High-Performance Ceramic Package. *IBM Journal of Research and Development*, 26:30–36, 1982.
- [4] J.-D. Cho. *High-Performance Physical Design in Multilayer Packages*. PhD thesis, Northwestern University, Evanston, Illinois, June 1993.
- [5] J. Darnauer and W. W. Dai. Fast Pad Redistribution for Periphery-IO to Area-IO. In *Proc. Multichip Module Conf.*, pages 38–43, Santa Cruz, CA, March 1994.
- [6] P. Dehkordi and D. Bouldin. Design for Packageability: Early Consideration of Packaging from a VLSI Designer's Viewpoint. *Computer*, 1:76–81, Apr. 1993.

- [7] P. Dehkordi and D. Bouldin. Design for Packageability: The Impact of Bonding Technology on the Size and Layout of VLSI Dies. In *Proc. IEEE Multichip Module Conf.*, pages 153–159, 1993.
- [8] P. Dehkordi, K. Ramamurthi, D. Bouldin, H. Davidson, and P. Sandborn. Impact of Packaging Technology on System Partitioning: A Case Study. In *Proc. Multichip Module Conf.*, pages 144–149, Santa Cruz, CA, January 1995.
- [9] G. T. Hamachi. An Obstacle-Avoiding Router for Custom VLSI. Technical Report UCB/CSD 86/291, Univ. of California, Berkeley, CA, April 1986.
- [10] W. Huang and J. Castro. CBGA Package Design for C4 PowerPC Microprocessor Chips: Trade-off between Substrate Routability and Performance. In *IEEE 44th Electronic Components & Technology Conf.*, pages 88–93, Washington, D. C., May 1994.
- [11] G. Kromann, D. Gerke, and W. Huang. A Hi-Density C4/CBGA Interconnect Technology for a CMOS Microprocessor. In *Proc. IEEE 44th Electronic Components & Technology Conf.*, pages 22–28, Washington, D. C., May 1994.
- [12] C. Y. Lee. An Algorithm for Path Connections and Its Applications. *IRE Trans. Electronic Computers*, pages 246–265, September 1961.
- [13] K. C. Norris and A. H. Landzberg. Reliability of Controlled Collapse Interconnections. *IBM Journal of Research and Development*, 13:266–271, 1969.
- [14] P. Sandborn, M. Abadir, and C. Murphy. The Tradeoff Between Peripheral and Area Array Bonding of Components in Multichip Modules. *IEEE Trans. on Components, Packaging and Mfg. Tech. - Part A*, 17(2):249–256, June 1994.
- [15] R. R. Tummala and J. Knickerbocker. Advanced Cofired Multichip Technology at IBM. In *Proc. IEPS.*, San Diego, CA, September 1991.
- [16] R. R. Tummala and E. J. Rymaszewski, editors. *Microelectronic Packaging Handbook*. Van Nostrand Reinhold, 1989.
- [17] M.-F. Yu and W. W.-M. Dai. Single-Layer Fanout Routing and Routability Analysis for Ball Grid Arrays. Technical Report UCSL-CRL-95-18, Univ. of California, Santa Cruz, CA., April 1995.

Dehkordi, P., Ramamurthi, K., Bouldin, D. and H. Davidson,
“Determination of Optimum Area-Array Bond Pitch for MCM Systems” ,
Proceedings of 1997 IEEE Multi-Chip Module Conference, Santa Cruz, CA, February 4-5, 1997.

Determination of Area-Array Bond Pitch for Optimum MCM Systems: A Case Study

Peyman Dehkordi, Karthi Ramamurthi, Donald Bouldin¹
and Howard Davidson²

Abstract

Microelectronics system designers need to understand and evaluate the impact of advanced packaging parameters which have become an integral part of microelectronics systems. This paper evaluates the impact of bond pitch for a flipchip multichip system through a case study. The SUN MicroSparc CPU was used as a representative of a large design where the design has to be partitioned and interconnected using MCM technology. Early analysis techniques were used to analyze the design for various pitches ranging from 150 to 400 micron in 50 micron increments. Results suggest that various bond pitches affect the system cost/performance and there is a minimum pitch at which lowering the pitch will degrade the cost/performance metrics.

Introduction

Optimization of microelectronics systems continue to be a difficult task since it involves a number of different disciplines. Since advanced packaging technologies are an integral part of high performance microelectronics systems, the optimization process becomes even more difficult. The system designer is now faced with dealing with the packaging issues as well as other traditional issues.

Traditionally, packaging selection and evaluation is done by the package designer within the limit of the packaging technology itself. However, some of these parameters have an impact at other phases of the design which may impact the system optimization. Therefore, it is important to identify some of these parameters and evaluate their impact for a given application at the system level to obtain optimum system design.

Bond pitch of a flipchip/MCM system is one of the important packaging parameters which can impact other

phases of the design (i.e. IC design and partitioning). This paper will evaluate the impact of the bond pitch at the system through a case study. In the following sections, the relationship between the bond pitch and other system parameters will be identified. The SUN MicroSparc CPU will be presented as a case study along with the results and conclusions.

Flipchip Bonding Technology

Area array bonding technologies such as flip-chip (these two names are interchangeably used throughout this paper) are gaining more popularity with the advancement in packaging technologies. Flipchip technology is expected to grow at a compound annual growth rate of 38% through the year 2001 [1]. This technology offers higher I/O count for a given die size as compared to the conventional wire-bond technology. This is due to the placement of the I/O pads on the area of the die rather than just on the periphery as shown in Figure 1. Therefore, the total I/O count is a function of the square of the die size and the bond pitch.

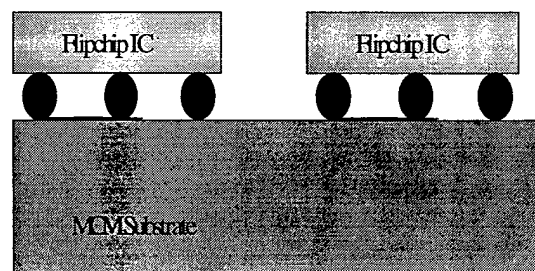


Figure 1. Flipchip Dies on MCM Substrate

¹ University of Tennessee, Knoxville, TN.

² Sun Microsystems, Mountain View, CA.

This work was sponsored in part by ARPA under grant DAAH04-94-G0004

Theoretically, from an IC point of view, decreasing the bond pitch (assuming manufacturing is possible) will result in a higher number of I/O for a fixed die area. However, from the substrate designer point of view, one has to be concerned with the "escape routing" of these pads as shown in Figure 2. In this paper, escape routing is referred to the routing of the area-array I/O pads from the area under/surrounding the die to the vias (for redistribution to other layers) or the top layer routing on the substrate. As explained in [2], the upper bound for the number of I/Os which can be placed on a die and escape routed is a function of the die size (determined by the IC technology), bond pitch (determined by the bonding technology), and the wiring pitch on the substrate (determined by the substrate technology). This upper bond density (number of I/O per unit area) is provided by [2]:

$$N = \frac{b^2}{4} \left(-1 - \sqrt{1 + \frac{4}{b}} \right)^2 \quad (1)$$

$$b = - \frac{4(P_{top} + 1)(2 + S:G) - 8}{S:G} \quad (2)$$

Where:

N is the total number of I/O that can be escape routed

P_{top} is number of tracks allowed between I/O pads

$S:G$ is the signal to ground ratio assuming equal number of power and ground

IC designers are usually faced with a high I/O requirement for their single chip designs. High I/O also impacts the partitioning boundaries of a multichip design (i.e. large design into smaller designs). Area-array bonding can offer a high I/O if the chip is designed specifically for area-array technologies (i.e. not a re-distribution of the periphery I/Os)[3]. For a globally-optimum system design, it is important to consider not only the I/O requirements at the IC level but also the escape routing requirements at the package level which in turn drives the partitioning boundaries in a multichip design. This requires understanding the interaction between the IC and the package at the system level and how to optimize the system accordingly.

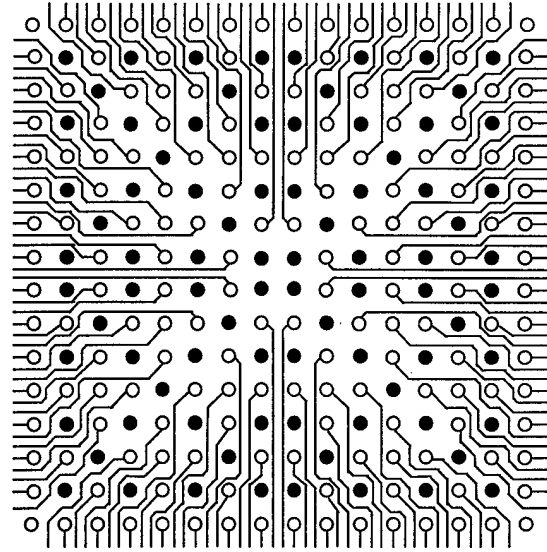


Figure 2. Possible Escape Routing from a uniform 16x16 Area Array Bond Pads. Shaded Points are Power and Ground I/Os [2].

Case Study and Results

In our previous work [4], we have proposed the concept of design for packageability (DFP) in which the packaging issues are considered at an early stage of the design for an optimum MCM solution. This concept was used to predict the cost/performance for a large design (SUN MicroSparc CPU) which was partitioned into smaller area-array dies and re-interconnected on an MCM-D[5]. This paper extends the previous study by considering the impact of the bonding pitch for the MicroSparc application.

We evaluated a multichip version of the MicroSparc CPU for flipchip bond pitch varying from 150 to 400 microns with 50 micron increments for an MCM-D technology. The MCM-D is assumed to be the MicroModule D500 merged via process[6]. A cost optimized partitioner was used to identify the partition with the lowest system cost as described in [4] (i.e. system cost = cost of IC + cost of substrate/package + cost of testing). The relative cost, size, noise, and power analysis are shown in Figures 3-7.

It is intuitively expected to have a lower system cost for lower bond pitch. This is in agreement with the cost results for 250-400 micron pitch. The reason for this trend is because lower pitch results in a higher I/O which in turn relaxes the I/O constraints for the partitioning. This allows the designs to be partitioned into smaller sets

of dies which may have high I/O count. Interestingly, the cost increases as the pitch changes from 250 to 200 micron. A careful investigation reveals that the number of tracks between the pads (i.e. P_{top} in equation 2) drops from three to two for smaller pitch (assuming the substrate technology remains the same). Therefore, when the escape routing is considered, smaller P_{top} results in smaller I/O density which will constraint the partitioning from generating a lower cost partition (i.e. the number of I/Os which can be **placed** on the **die** and **routed** on the **substrate** is limited by the escape routing rather than the size of the die). For this case study and given MCM-D technology, pad pitch of 150 micron does not allow partition of the design into a meaningful set of smaller ICs.

As the bond pitch is increased, there will be more tracks allowed between the bond pads, relaxing the escape routing on the substrate. However, if the die is I/O limited, the die size will be increased resulting in a higher system cost and size. If the bond pitch is decreased, it will increase the difficulty of escape routing and limit the number of I/Os on the die. This in turn will not allow lower cost partitions based on higher I/O requirements. As the results suggest, 250 micron pitch is the lowest cost system since it balances two opposing factors: the escape routing difficulty and lower bond pitch.

As Figure 4 suggests, the system size seems to be rather flat with gradual increase in size as the pitch decreases. Smaller pitch will allow the partitioner to create smaller ICs with higher I/Os (as long as I/Os are not escape routed limited). Since there is a fixed overhead associated with each IC (such as minimum chip-to-chip spacing), the partitioning process tends to increase the overall system size.

Figures 5 and 6 show the system power consumption and relative measure of the simultaneous switching noise. Both of these metrics are directly related to the total number of I/O in the system. The lowest cost case (250 micron) is also the most power consuming and contains the highest noise level. This is because the partitioner has generated smaller sets of dies with high I/O counts.

Conclusions

Impact of bond pitch of a partitioned multichip flipchip MCM-D system has been presented for pitches varying from 150 to 400 micron. A combination of the substrate trace pitch along with bonding pitch can affect the system partitioning and therefore system cost/performance. Intuition suggest that smaller bond pitch results in a lower cost system. For this case study and given cost model, the optimum pitch is at 250 micron. Reducing the pitch below this value will not improve the system cost/performance due to the difficulties in escape routing. A different cost model or a different application may result in a different optimum bond pitch.

Acknowledgments

The MSDA (Multichip System Design Advisor) tool has been extensively used throughout this work. The authors gratefully acknowledge the contribution of the MSDA tool which was originally developed by Peter Sandborn at MCC and is now being commercialized through Savantage, Inc.

References

- [1] World Wide Web home page of FlipChip Technologies, August, 1996, <http://www.flipchip.com>
- [2] Sandborn, P. and H. Moreno, *Conceptual Design of Multichip Modules and Systems*, Kluwer Academic Publishers, 1994.
- [3] Dehkordi, P., Tan, C., Bouldin, D. "Intrinsic Area Array ICs: What, Why, and How", *Proceedings of the MCM Conference*, February 1997.
- [4] Dehkordi, P., Ramamurthi, K., Bouldin, D., Davidson, H. and P. Sandborn, "Impact of Packaging Technology on System Partitioning: A Case Study", *Proceedings of the MCM Conference*, March, 1995.
- [5] Dehkordi, P., Ramamurthi, K., Bouldin, D., Davidson, H., "Early Cost/Performance Cache Analysis of a Split MCM-Based MicroSparc CPU", *Proceedings of the MCM Conference*, February, 1996.
- [6] MicroModule System Corp., *MCM-D Technology Kit, Rev. 1.01*, MicroModule System Corp., Cupertino, CA, 1994.

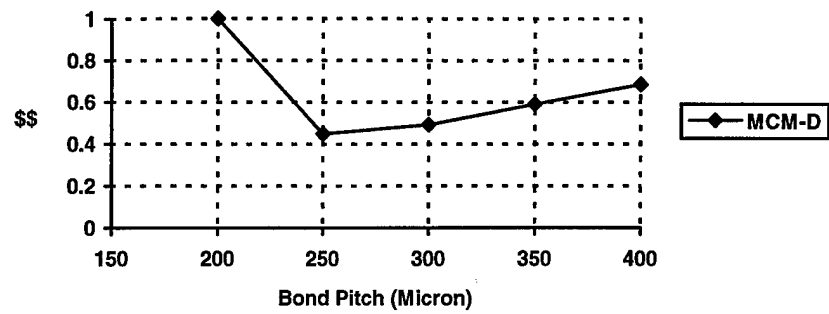


Figure 3. The Relative Comparisons of System Cost for Various Bond Pitch

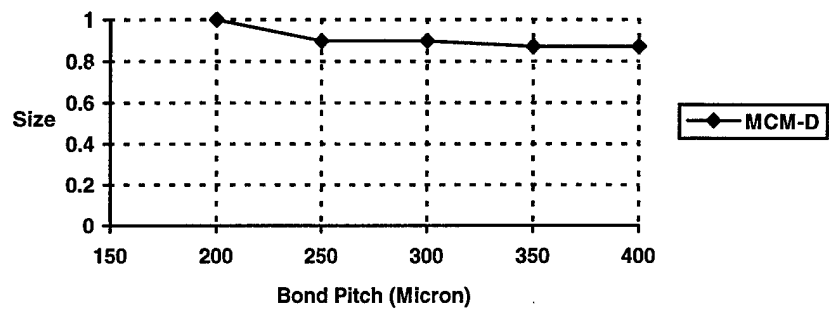


Figure 4. The Relative Comparisons of System Size for Various Bond Pitch

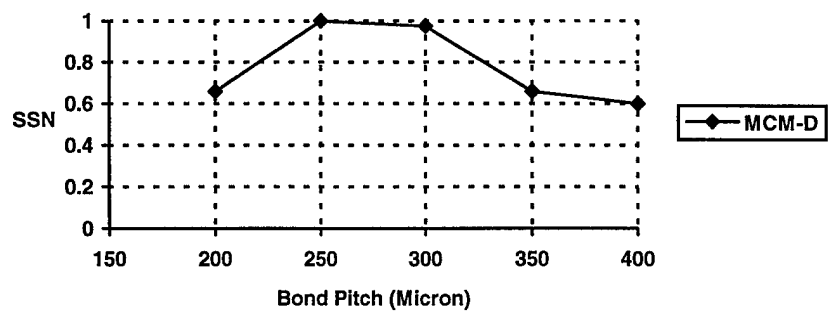


Figure 5. The Relative Comparisons of System Noise for Various Bond Pitch

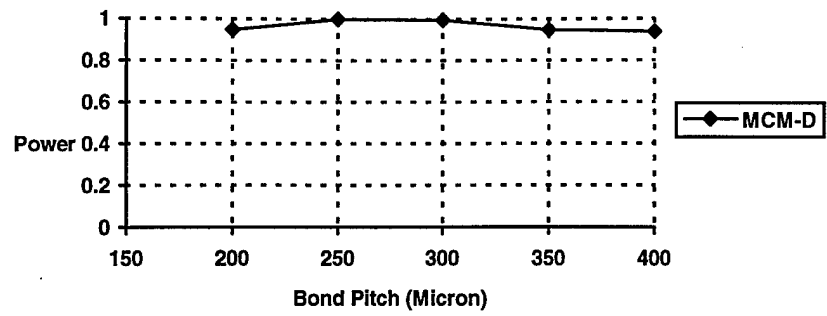


Figure 6. The Relative Comparisons of System Power for Various Bond Pitch

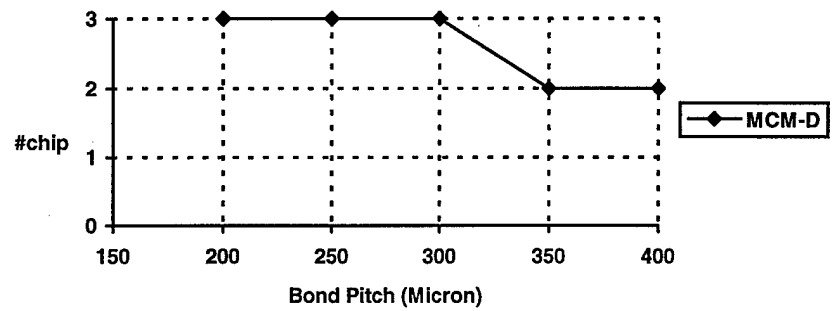


Figure 7. The Number of ICs in the Partition for Various Bond Pitch

Tan, C., Bouldin, D. and P. Dehkordi,
“An Intrinsic Area-Array Pad Router for ICs” ,
Proceedings of Tenth Annual IEEE International ASIC Conference , pp. 265-269,
Portland, OR, September 7-10, 1997.

AN INTRINSIC AREA-ARRAY PAD ROUTER FOR ICS

Chandra Tan, Donald Bouldin, and Peyman Dehkordi

Department of Electrical Engineering

The University of Tennessee

Knoxville, TN 37996-2100

chandra,bouldin,dehkordi@microsys1.engr.utk.edu

ABSTRACT

Arranging I/Os in a matrix array over the core circuitry of an IC generally provides 5-10 times more I/Os than the traditional method of restricting pads to the periphery. This approach also minimizes overall die size. In this paper we describe the development of a new area-array pad router which differs from other approaches in that no additional metal layer is added (unless needed) and no redistribution is required. We describe the design implementation of this technique and show the results of applying this router on designs requiring 112, 298, 414 and 485 I/Os.

I. INTRODUCTION

As the functions of ICs have been remarkably enhanced, the number of pads for interconnections on a chip has increased significantly in recent years. This reason leads to the realization that a denser area technique for chip connection will have to be employed. One type of chip connection is the area-array pad technique which generally uses the flip-chip assembly method. Known also as the C4 process (Controlled Collapse Chip Connection), flip-chip bonding was developed by IBM in 1964 and has been used for over 30 years to assemble IBM's mainframe computer modules [12]. Flip-chip gives the highest chip density of any packaging method to support pad-limited IC designs. Instead of placing the chips in space-wasting individual packages, flip-chip places them face down onto matching connections on a substrate or board by means of solder pads or bumps.

Presently, there are two major approaches for designing an area-array IC: **intrinsic** and **extrinsic**. An intrinsic area-array IC is one especially designed and laid out for area-array bonding. Several approaches for designing an intrinsic area-array IC have been reported recently [4, 6, 8]. An extrinsic area-array IC is one originally designed and laid out for periphery bonding but converted for area-array bonding by means of a redistribution layer. Recently, [2, 9] reported techniques for designing extrinsic area-array ICs.

The authors gratefully acknowledge the support of DARPA grant DAAH04-94-G0004

In our previous paper presented in [4], we showed why an intrinsic approach to design area-array IC was needed and showed the impact of this approach on the design at the IC level as well as at the system level. The purpose of this paper is to present the design implementation of this approach. We will also show the results of applying our technique to three area-array IC designs.

II. RELATED WORKS

Cascade Design Automation recently reported a preliminary version of a CAD tool for designing area-array ICs [6]. This tool consists of an area-pad power analyzer, an area-pad floorplanner and an area-pad router. The paper presented by Kiamilev et. al. [8] demonstrated three methods of designing an intrinsic area-array IC. The first method can only be used to design some special ICs with an array structure. The second method is a general approach that can be used to design any IC in which the area-distributed padframe is integrated directly over active circuitry of the IC. However, the placement and routing of the area pads must be done manually. In their final method, the I/O pad circuits were placed and optimized separately from the core of the chip. The area pads are placed directly over their corresponding I/O pad circuits to form a two-dimensional array structure called the pad interface module. This module was then placed in the center of the IC. This method can be used for large designs because it is compatible with current VLSI CAD tools, thus enabling the automation of placement and routing tasks. However, the density of this type of design is much lower than the design achieved using the second method.

In this paper, we present the development of an area-array pad router which automates the placement and routing of the area-array pads on the IC. This is a post-processing tool which allows IC layouts generated by any other tool to be processed for area-array I/O pad placement and routing. This tool is different from a previously reported tool [2] since ours will place and route the pads using existing layers on the IC (if possible) before adding a new redistribution layer. Our method is also different from that presented

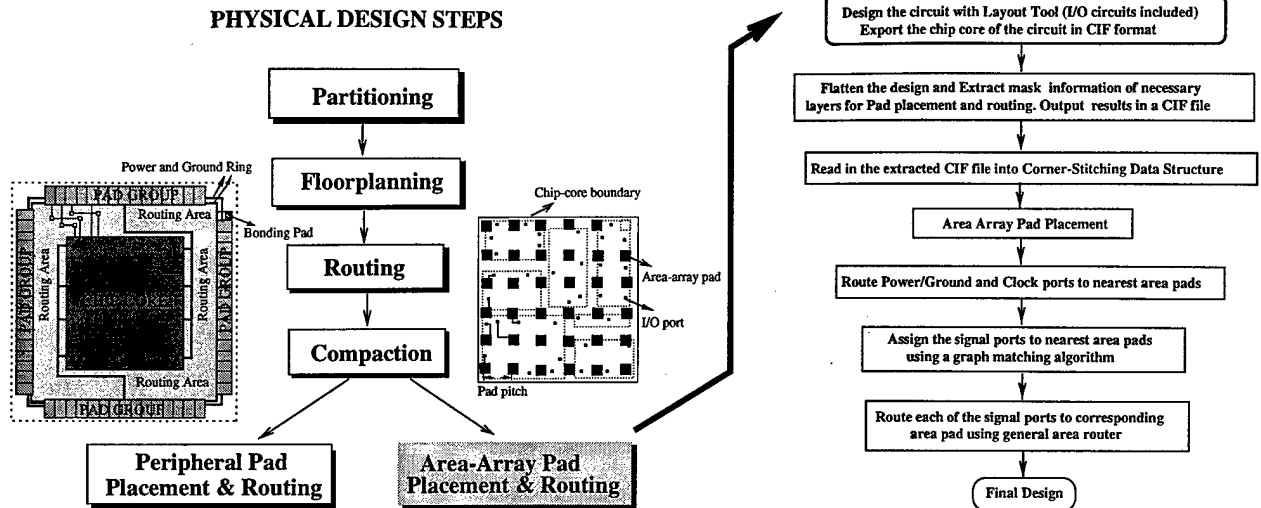


Figure 1: Physical Design Flow of Area-Array ICs.

in [6] since we place the area-array pads on the pre-existing top metal layer. Routing of the I/O ports to these area pads is performed on the top few metal layers specified by the designer. Our approach is similar to the second method presented in [8]. However, in our approach, the placement and routing of area-array pads are performed automatically.

III. PHYSICAL DESIGN FLOW OF AREA-ARRAY ICs

Physical design of a conventional peripheral IC consists of partitioning, floorplanning, routing, compaction, and pad placement and routing steps. In designing an intrinsic area-array IC, we adapted most of these procedures as shown in Figure 1. The framework for designing an intrinsic area-array IC is divided into several substeps: design guideline definition, data preparation, pad placement, pad assignment, pad routing, and output padframe generation. The detailed description of these steps is given in [11]. We modified some of the sub-steps of conventional physical design to incorporate some design guidelines to facilitate the placement of the area-array pads and the routing of the connections between the I/O ports and the area-array pads. The placement and routing of the pads are performed in the final stage of the design steps.

Before proceeding to the pad placement and routing, the chip core layout is flattened to extract the geometric information on the top few layers, the pad keepout areas, the locations of I/O terminals, and the boundary of the core circuits. Utilizing this extracted information, the pad placement algorithm calculates

the center of the chip core and then uses this center as a reference point. The router then generates a set of the potential area-array bonding pads on the top metal layer satisfying the user-specified bond pitch. Using a weighted bipartite matching algorithm developed by Jonker and Volgenant [7], the I/O ports are assigned to area pads with the objective of minimizing the total wire length. This assignment generates a two-terminal netlist to be input into the pad routing routine to make the complete wiring. In the routing phase, the power/ground and clock I/O pads are routed first before the signal I/Os. Before routing the signal I/Os, net ordering is performed on the netlist to give the I/O port closest to the center of chip core boundary a higher priority. The general area router adapted in this paper is based on a maze router [1, 5] with the A* search principle and the corner-stitched data structure [10]. If the routing fails, we record this net into an array. If the routing of the net is successful, a glass cut and a label with suffix "pad" are added to the targeted area pad. When all the nets have been processed, the unrouted nets are routed manually with computer assistance. Our tool provides an option to find the location of the unrouted port and its assigned area pad by zooming into the area containing the I/O port and highlighting both the I/O port and the area pad. When all the nets are routed, the unused area pads located at the periphery of the pad array may be assigned as dummy pads. The remaining unused area pads are erased from the layout. The final padframe (area-array pads along with the wire connections) is saved into a CIF file to be appended to the CIF file of the original design to form an area-array IC.

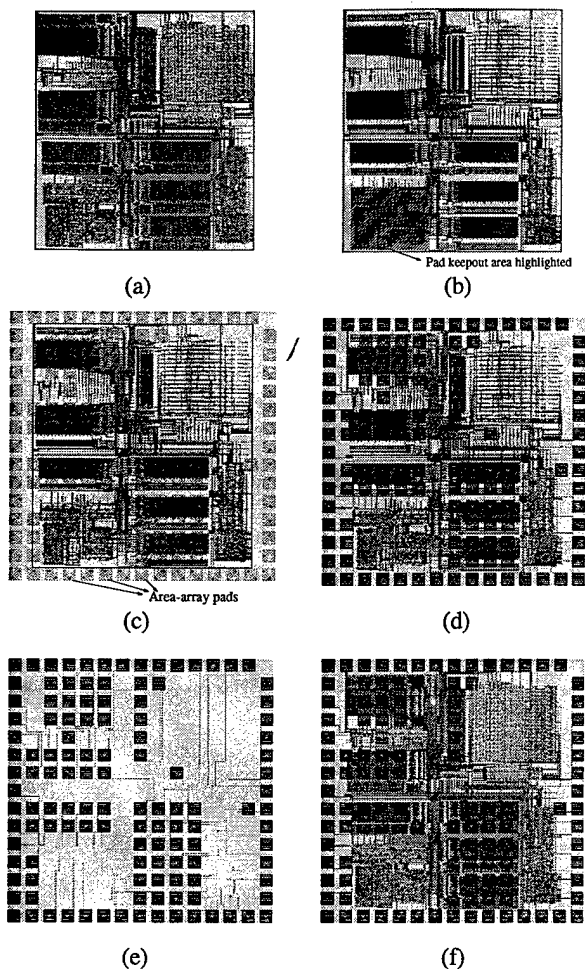


Figure 2: The "Quantizer" chip used as example to illustrate the implementation of our tool.

IV. IMPLEMENTATION

The framework for intrinsic area-array IC placement and routing has been implemented within the Berkeley *Magic* CAD Tool in C and run on a SUN SparcStation. Next we will describe the various steps of our tool in more detail. Given the mask geometries of the chip core stored in CIF (Caltech Intermediate Form), our tool generates the area-array padframe of the IC. In Figure 2, each of the substeps in the implementation of our tool is illustrated. The "Quantizer" chip shown in Figure 2(a) was taken from a Cascade Design Automation tutorial example to serve as our first implementation example.

Before the tool proceeds to the pad placement, pad assignment, and pad routing steps, *cifextr* is used to flatten the chip core layout to extract the geometric information on the top few layers (the number of layers is specified by the designer), the pad keepout areas, the locations of I/O terminals, and the boundary of the core circuits. Figure 2(b) shows the layout of the top two metal layers extracted using *cifextr* from the

core design of the Quantizer.

The pad placement algorithm utilizes two of the database routines in *Magic*, the point finding and the area enumeration routines, to place the pads on the top metal layer one at a time. The point finding routine is used to locate the potential pad site and the area enumeration routine is used as a search technique. Figure 2(c) shows the result of applying the pad placement algorithm. The pad size is $80\ \mu\text{m}$ and the pitch is $120\ \mu\text{m}$. The number of pads placed is 124 which is 10% greater than the number of I/O ports, 112.

After performing the pad assignment to generate a two-terminal netlist, our tool performed the routing of the nets based on the net ordering described in previous section. In this case, only two of the nets failed to be routed automatically. They were routed manually with the assistance provided in our tool. Figure 2(d) shows the result of the routing on Quantizer chip.

When placement and routing of area-array pads are completed, our tool generates an area-array padframe by erasing all the layout information stored in the extracted CIF file and keeping intact the area-array pads and their routed wires to the I/O ports. This area-array padframe is then stored in a CIF file to be appended to the CIF file of original core design to form a complete area-array IC. Figure 2(e) shows the area-array padframe of the Quantizer chip and Figure 2(f) shows a complete area-array Quantizer IC. The chip has 112 bonding pads of size of $80\ \mu\text{m}$ and pitch of $120\ \mu\text{m}$. The total size of this version is $1.76 \times 1.76\ \text{mm}$.

For more demonstration of the methodology of designing an intrinsic area-array IC, we conducted the experiments on the design of 3 partitioned chips from the SUN MicroSparc design resulted from the study conducted in [3]. These three partitioned chips are denoted as ChipA, ChipB, and ChipC.

All three chips were resynthesized and laid out using the EPOCH physical design automation tool. The core circuits were exported to CIF files. The area-array pads were placed on metal-3 layer and the pad routing was done on the metal-2 and metal-3 layers. For ChipA, the routing of the area pads took approximately 8 hours on an UltraSparc with 512 Megabytes of internal memory. There were 46 nets which were not routed automatically. It took another hour or so to manually route these nets using area router with the assistance provided in our method. The reason it took so long to route the I/O ports to their assigned area pads was due to the clustering of the I/O port locations at certain regions in the core area. In this case most of the I/O ports were located at the bottom-left of the core area. Hence, some of the I/Os were assigned pads that were located at the top half of the core area. Therefore, the routing of these net-

s consumed a lot of time and memory space. If we could scatter the I/O ports all over the core area, the routing would be faster and the number of unrouted nets would be reduced. For ChipB, the routing of the area pads took approximately 6 hours for the same reason explained in the ChipA case. There were 16 nets which had to be routed manually with the assistance provided in our method. Finally, for ChipC, the routing of the area pads took approximately 3 hour. There were 16 nets which had to be routed manually.

The final results for these three ICs for both peripheral and area-array pads are shown in Figure 3. Comparison of chip sizes for these ICs area summarized in Table 1. We purposely used different pad size and pitches for the above three different designs to show the flexibility of our tool. From the results shown in this table, we can see that a significant reduction in chip size can be achieved by converting the peripheral IC with high I/O count to an intrinsic area-array IC (ChipA and ChipC cases).

V. SUMMARY & CONCLUSIONS

The framework for designing an intrinsic area-array IC has been developed in this paper. This framework is divided into several substeps: design guideline definition, data preparation, pad placement, pad assignment, pad routing, and output padframe generation. For each of these substeps, the solution method and its implementation have been presented.

The method developed in this paper has several advantages over some previous approaches: there is no need for extra layers for placing the area-array pads and redistribution. In fact, the placement and routing of area pads are performed on the pre-existing metal layers. The resulting area-array IC should be smaller in size and cheaper in cost for high I/O count. Our method can also serve as a postprocessor to any IC physical design tool for designing area-array ICs. In our implementation, we used the EPOCH to layout the core circuits of the design following the guidelines provided in our method. The layout was then exported to a CIF file to be imported into our tool. An area-array padframe was generated to be appended to the core design to form an intrinsic area-array IC.

Placing and routing area-array pads by hand will be virtually impossible and very time-consuming as the number of I/Os become large. As flip-chip design approaches with high I/O count become more common in the future, there will be a greater demand to place and route area-array pads for hundreds of I/Os automatically. The framework presented in this paper for designing an intrinsic area-array IC will be useful in this case.

VI. ACKNOWLEDGEMENT

The EPOCH Physical Design tool by Cascade Design Automation has been extensively used for this research. The authors gratefully acknowledge the donation of this tool to the University of Tennessee. The authors also acknowledge the contribution of the *Magic* layout tool to this work.

VII. REFERENCES

- [1] M. H. Arnold and W. S. Scott. An Interactive Maze Router with Hints. In *Proc. 25th Design Automation Conf.*, pages 672-676, Anaheim, CA, June 1988.
- [2] J. Darnauer and W. W. Dai. Fast Pad Redistribution for Periphery-IO to Area-IO. In *Proc. Multichip Module Conf.*, pages 38-43, Santa Cruz, CA, March 1994.
- [3] P. Dehkordi, K. Ramamurthi, D. Bouldin, H. Davidson, and P. Sandborn. Impact of Packaging Technology on System Partitioning: A Case Study. In *Proc. Multichip Module Conf.*, pages 144-149, Santa Cruz, CA, 1995.
- [4] P. Dehkordi, C. Tan, and D. Bouldin. Intrinsic Area-Array ICs: What, Why, and How. In *Proc. Multichip Module Conf.*, pages 120-124, Santa Cruz, CA, February 1997.
- [5] J. Dion and L. M. Monier. Contour: A Tile-based Gridless Routing. Technical Report WRL Research Report 95/3, Digital Equipment Western Research Laboratory, March, 1995.
- [6] R. Farbarik, X. Liu, M. Rossman, P. Parakh, T. Basso, and R. Brown. CAD Tools for Area-Distributed I/O Pad Packaging. In *Proc. Multichip Module Conf.*, pages 125-129, Santa Cruz, CA, February 1997.
- [7] R. Jonker and A. Volgenant. A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems". *Computing*, 38:325-340, 1987.
- [8] F. E. Kiamilev, A. V. Krishnamoorthy, J. R. R. G. Rozier, G. F. Aplin, C. D. Hull, R. Farbarik, and R. E. Oettel. Design of ICs for Flip-Chip Integration with Optoelectronic Device Arrays. In *Proc. Multichip Module Conf.*, pages 163-167, Santa Cruz, CA, February 1997.
- [9] G. Kromann, D. Gerke, and W. Huang. A Hi-Density C4/CBGA Interconnect Technology for a CMOS Microprocessor. In *Proc. IEEE 44th Electronic Components & Technology Conf.*, pages 22-28, Washington, D. C., May 1994.
- [10] J. K. Ousterhout. Corner Stitching: A Data Structuring Technique for VLSI Layout Tools. *IEEE Trans. on Computer-Aided Design*, CAD-3(1):87-100, 1984.
- [11] C. Tan, D. Bouldin, and P. Dehkordi. Design Implementation of Intrinsic Array Array ICs. To appear in *Proc. Advanced Research in VLSI Conf.*, Ann Arbor, Michigan, September 1997.
- [12] R. R. Tummala and E. J. Rymaszewski, editors. *Microelectronic Packaging Handbook*. Van Nostrand Reinhold, 1989.

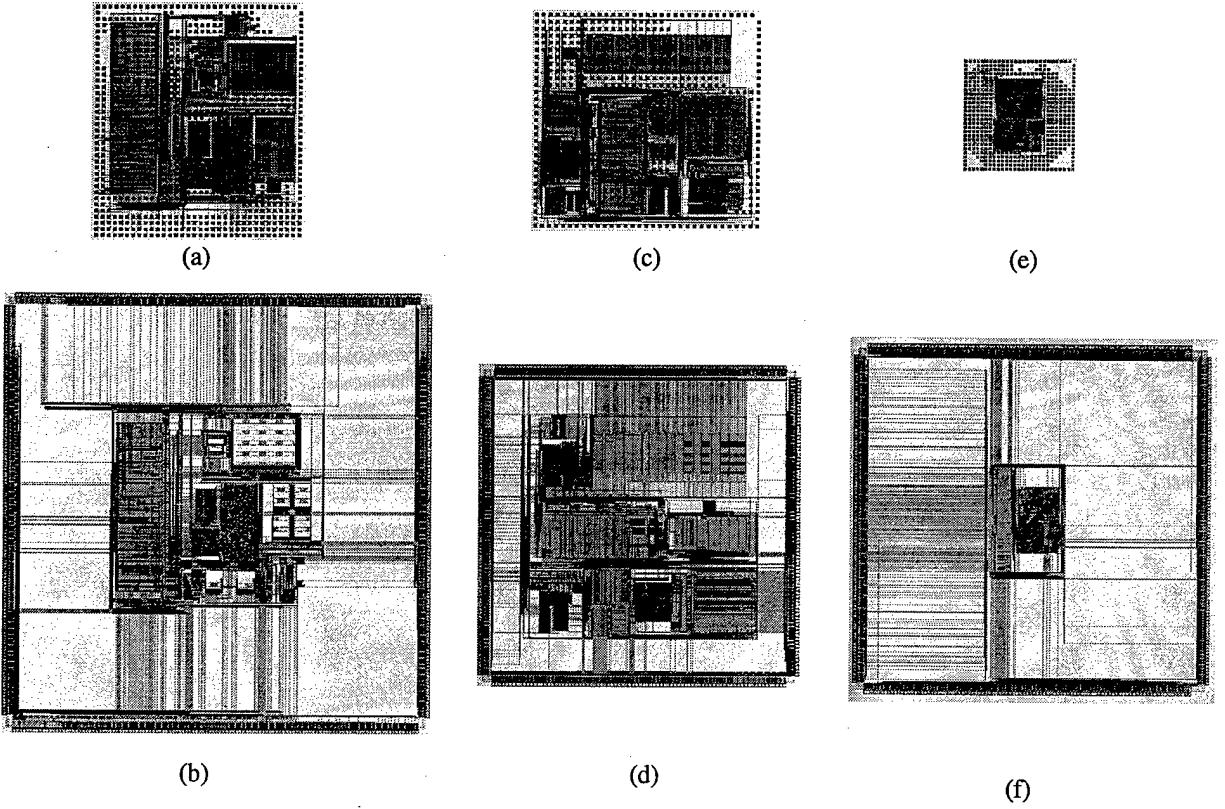


Figure 3: Area-array Vs peripheral ICs: (a), (c), and (e) Area-array IC and (b), (d), and (f) Peripheral IC of ChipA, ChipB, and ChipC, respectively.

Table 1: Comparison of chip sizes for area-array and peripheral IC for ChipA, ChipB, and ChipC.

Chip	# of I/Os	Peripheral IC			Area-array IC		
		pad size (μm)	pad pitch (μm)	chip size (μm^2)	pad size (μm)	pad pitch (μm)	chip size (μm^2)
A	485	100×100	110	20.18e+7	100×100	200	5.025e+7
B	298	100×100	110	9.9e+7	100×100	225	5.0e+7
C	414	100×100	110	12.8e+7	80×80	120	1.18e+7

Tan, C., Bouldin, D. and P. Dehkordi,
“Design Implementation of Intrinsic Area Array ICs” ,
Proceedings of 1997 Advanced Research in Very Large Scale Integration Conference, Ann Arbor, MI,
September 15-16, 1997.

Design Implementation of Intrinsic Area Array ICs

Chandra Tan, Donald Bouldin, and Peyman Dehkordi*

Department of Electrical Engineering

The University of Tennessee

Knoxville, TN 37996-2100

{chandra,bouldin,dehkordi@microsys1.engr.utk.edu}

Abstract

Arranging I/O in a matrix array over the core circuitry of an IC generally provides 5-10 times more I/O than the traditional method of restricting pads to the periphery. This approach also minimizes overall die size. This method was pioneered by IBM over thirty years ago and has recently become attractive for new designs requiring several hundred I/O. In this paper we describe the development of a new area-array pad router which differs from other approaches in that no additional metal layer is added (unless needed) and no redistribution is required. We describe the design guideline definition, data preparation, pad placement, pad assignment, pad routing, and output padframe generation of this technique. The results of applying this router are shown for sample designs requiring 112, 298 and 485 I/Os.

1: Introduction

Area-array solder interconnections can be made using the flip-chip assembly technique. Known also as the C4 process (Controlled Collapse Chip Connection), flip-chip bonding was developed by IBM in 1964 and has been used for over 30 years to assemble IBM's mainframe computer modules [18]. Flip-chip gives the highest chip density of any packaging method to support pad-limited IC designs. Instead of placing the chips in space-wasting individual packages, flip-chip places them face down onto matching connections on a substrate or board by means of solder pads or bumps.

Recently, most military and commercial electronic products are evolving toward lower cost, smaller form factor, lower weight and higher performance. All of these improvements can be achieved at the chip level by eliminating the IC packages and transitioning from perimeter to area I/O.

Presently, there are two major approaches for designing an area-array IC: **intrinsic** and **extrinsic**. An intrinsic area-array IC is one specially designed and laid out for area-array bonding. Several approaches for designing an intrinsic area-array IC have been reported recently [5, 7, 10]. An extrinsic area-array IC is one originally designed and laid out for periphery bonding but has been converted for area-array bonding by means

*The authors gratefully acknowledge the support of DARPA grant DAAH04-94-G0004.

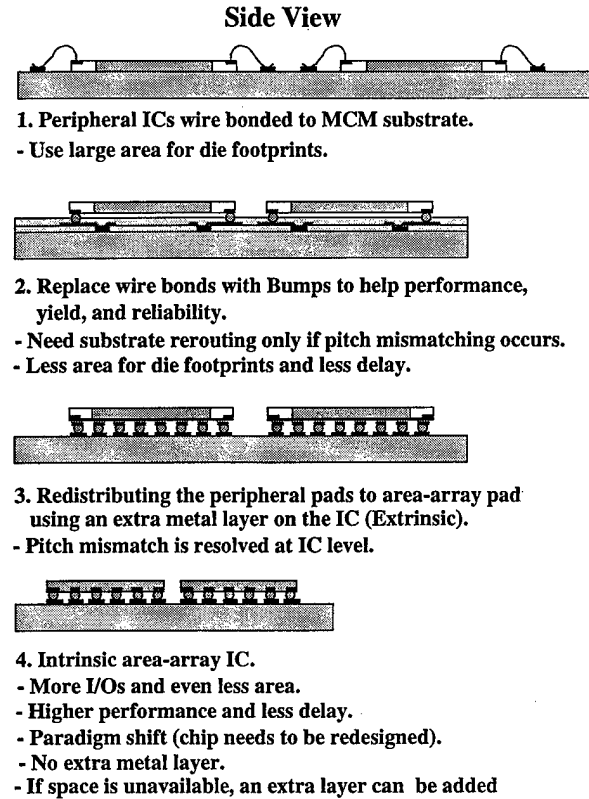


Figure 1. Motivation for Intrinsic Area-Array ICs.

of a redistribution layer. Recently there are some reports on converting the peripheral wire-bond IC to area-array flip-chip IC using a redistribution layer [3, 11].

In our previous paper presented in [5], we showed why an intrinsic approach to design area-array IC was needed and showed the impact of this approach on the design at the IC level as well as at the system level. Figure 1 summarizes the reason we need an intrinsic area-array IC and the improvement that can be achieved using this approach. The purpose of this paper is to present the design implementation of this approach. We will also show some area-array ICs designed using this approach.

2: Related Works

Cascade Design Automation recently reported a preliminary version of a CAD tool for designing area-array ICs [7]. This tool consists of an area-pad power analyzer, an area-pad floorplanner and an area-pad router. The paper presented by Kiamilev et. al. [10] demonstrated three methods of designing an intrinsic area-array IC. The first method can only be used to design some special ICs with an array structure. The second method is a general approach that can be used to design any IC in which the area-distributed padframe is integrated directly over active circuitry of the IC. However, the placement and routing of the area pads must be done manually. In their final method, the I/O pad circuits were placed and optimized separately from the core of the chip. The area pads are placed directly over their corresponding I/O pad circuits to form a two-dimensional array structure called

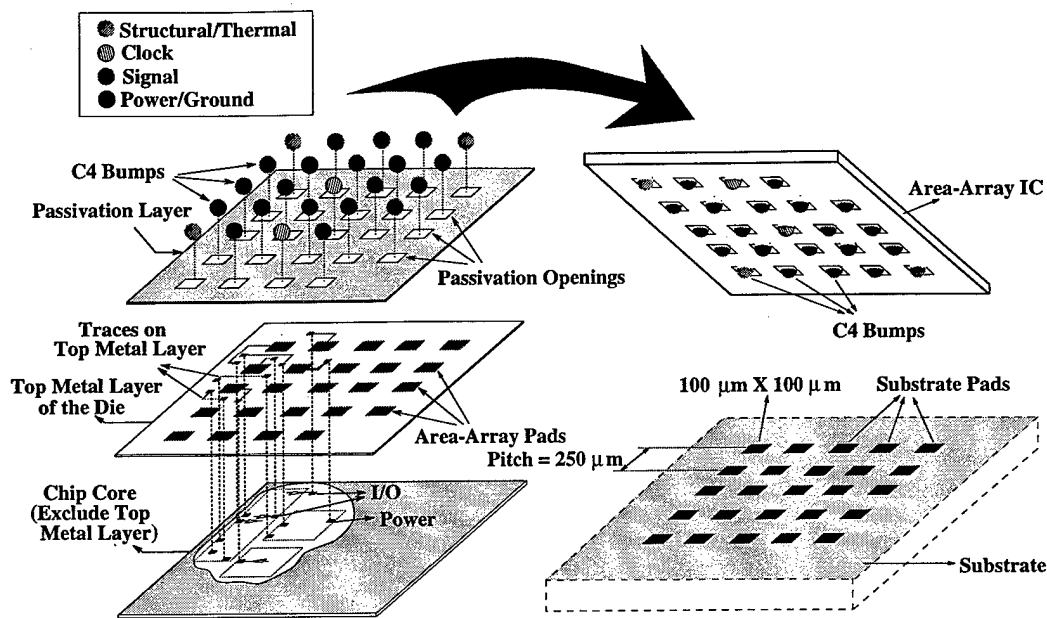


Figure 2. Intrinsic Area-Array IC Layout Problem Definition.

the pad interface module. This module is then placed in the center of the IC. This method can be used for large designs because it is compatible with current VLSI CAD tools, thus enabling the automation of placement and routing tasks. However, the density of this type of design is much lower than the design achieved using the second method.

2.1: Problem Definition

We present the development of an area-array pad router which automates the placement and routing of the area-array pads on the IC. This is a post-processing tool which allows IC layouts generated by any other tool to be processed for area-array I/O pad placement and routing. This tool is different from a previously reported tool [3] since ours will place and route the pads using existing layers on the IC (if possible) before adding a new redistribution layer. Our method is also different from that presented in [7] since we place the area-array pads on the top metal layer with some prerouted wires and routing of the I/O ports to these area pads is performed on the top few metal layers specified by the designer. Our approach is similar to the second method presented in [10]. However, in our approach, the placement and routing of area-array pads are performed automatically. Figure 2 shows some of the design steps for the intrinsic area-array IC layout problem.

Unlike the physical design of a periphery-bonded IC which places and routes the I/O cells (pads and I/O buffers) around the die after the active components of the chip have been laid out, the I/O buffers for area-array pads and the active components are laid out at the same time. The area-array bonding pads will then be placed on the top metal layer of the die and routed to their corresponding I/O buffers.

The layout problem of an intrinsic area-array IC can be stated as: Given a core portion of the chip which already contains the I/O buffers, place the possible uniformly spaced area-array pads on the top metal layer of the design (which may contain some prerouted wires and keepout areas) and route these pads to the I/O ports of the chip using a limited number of available connection layers, such that the design rules are obeyed and some other

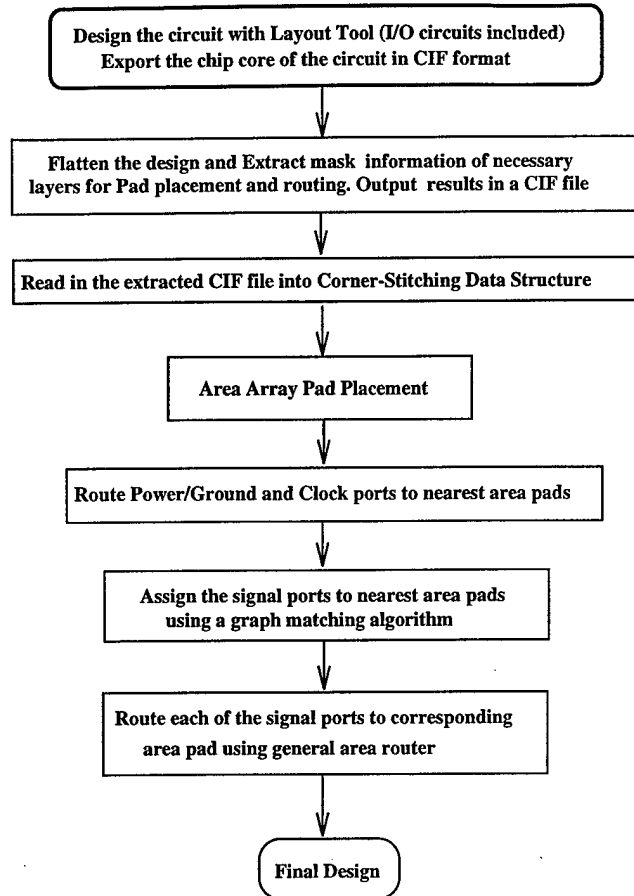


Figure 3. Intrinsic Area-Array Pad Router Design Flow.

objective functions are satisfied.

3: Physical Design Flow of Area-Array ICs

In designing the area-array IC, we adapted most of the procedures for designing conventional peripheral ICs. We modified some of these sub-steps to incorporate some design guidelines to facilitate the placement of the area-array pads and the routing of the connections between the I/O terminals and the area-array pads. The placement and routing of the pads performed in the final stage of the design steps. In other words, the algorithm, developed in this paper, can be used as a post-processor to existing VLSI layout tools to design an area-array IC.

The design flow of our approach is given in Figure 3. Given the core circuits designed following the design guidelines below, the process starts with the extraction of the top layer (more than one layer is also possible). Then the area-array pads are placed at potential sites on the top layer and each of the I/O ports is routed to the closest pad. After routing, a passivation layer is applied to insulate the lines. Vias are opened in the passivation layer to allow placement of solder balls for flip-chip attachment.

3.1: Design Guidelines

To facilitate the design of a VLSI chip with area-array bonding technology, the chip designer needs to follow these design guidelines:

1. The pad and the buffering circuitry may be separated. The I/O buffers need to be placed and routed to the active components in the chip core during the first three sub-steps of the physical design.
2. The I/O ports of these buffers must be labeled with a special attribute.
3. The I/O ports must contain one of the layers or contacts required in the pad routing stage in our tool.
4. Submodules in the design which prohibit the placement of the pads should be labeled as pad keepout areas.

3.2: Data Extraction

Most VLSI circuits are designed using a hierarchical approach. Cells in a hierarchical layout may contain mask information, or they may contain other cells, known as *subcells*. Because area-pad placement and routing require the layout to be represented in a single set of mask layers, we need to flatten the chip core layout and extract the number of specific routing layers, the pad keepout areas and chip core boundary as well as the I/O port sites. This step is done using a program called *cifextr* (a modified version of *cifflatten* developed at the University of California at Berkeley). The bounding box of each pad keepout area is stored on a special layer and the chip core boundary is marked at its lower-left and upper-right corners.

3.3: Pad Placement

The pad placement algorithm takes as input a layout information on extracted layers including prerouted wires, pad keepout areas, and the chip core boundary. It calculates the center of the chip core and then, using this center as a reference point, generates a set of possible C4 bonding sites based on the user-specified pad size and pad pitch. If the bonding site is inside the core boundary and not covered by any pad keepout areas, a searching box is created. This searching box is a square area created by adding a minimum spacing width to all four sides of the area pad to ensure that no spacing violations could ever be created when placing the pads. Using an area searching technique, a search is conducted for any prerouted wire under the area covered by this box. Also, a check is made to see whether this box is overlapping any keepout area. Next, a pad is placed at the center of this location if no wire is found and no overlapping occurred. If the bonding site is outside the core boundary, the bonding pad is placed directly at this site. This process is continued until the number of pads placed is 10% larger than the number of I/O ports. The pad location at the upper-right corner of this array is purposely omitted for identification of chip orientation. Figure 4 shows this pad placement method.

3.4: Pad Assignment

Some of the I/Os that need to be at certain predefined pad locations can be assigned by the designer manually. These assigned pairs are routed sequentially using the maze router with the I/O port as the source and the area pad as the target. In this case, the source is a point and the target is a square. The remaining I/Os are also routed sequentially using

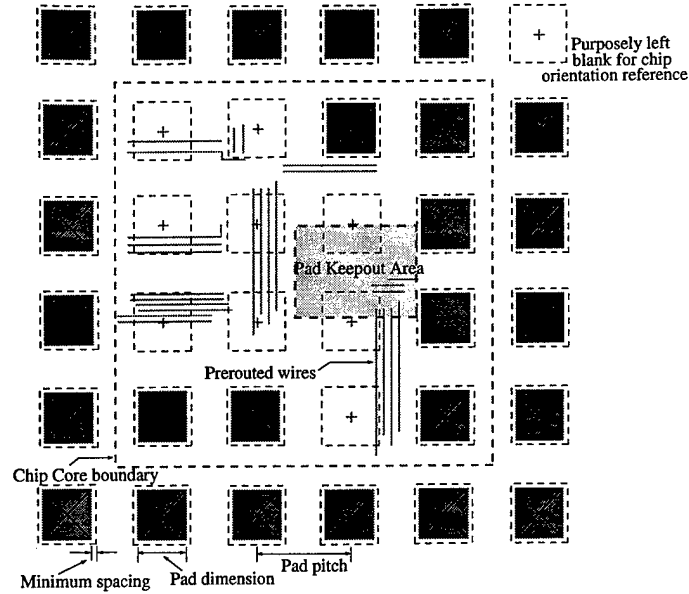


Figure 4. Pad Placement Strategy.

the maze router based on the A* search. The netlist is generated using a pad assignment method which minimizes the total Manhattan distance. This assignment problem can be formulated as a weighted bipartite matching problem. This problem has been studied extensively in the field of network optimization and operations research [15]. Some of the most efficient algorithms for solving this matching problem are based on the successive augmenting path method. The successive shortest augmenting path algorithm developed by Jonker and Volgenant [9] uses a label-setting algorithm to find the shortest augmenting path and has been adapted by us to solve the pad assignment problem.

3.5: Pad Routing

Pad routing is performed on the few top metal layers defined by the designer. Thus, the routing region is represented by different, arbitrary, rectilinear polygons on each of its multiple routing layers which may contain tens or even hundreds of arbitrary blockages due to the prerouted wires. In addition, the I/O ports may be positioned anywhere within the routing region or on its border. Because of these conditions, channel or switchbox routing methods cannot be applied since the usual channel or switchbox routing areas do not exist. Therefore, a general area router has been adapted by us to solve this pad routing problem.

The general area routing problem can be viewed as a collection of search problems. The individual search problems deal with finding a route, and laying down a net (wire), between each pair of I/O ports and area pads. There are many search algorithms for solving the routing problem [2, 12, 13, 16, 17]. The general area router adapted in this paper is based on a maze router [1, 6] with the A* search principle and the corner-stitched data structure [14]. The A* utilizes a heuristic function to guide its search towards the target. The heuristic function is a cost function from any point to the target. The cost can be an estimated distance from the point to the target. Let v be a node reached by a wavefront. The cost function is defined as:

$$f(v) = g(v) + h'(v)$$

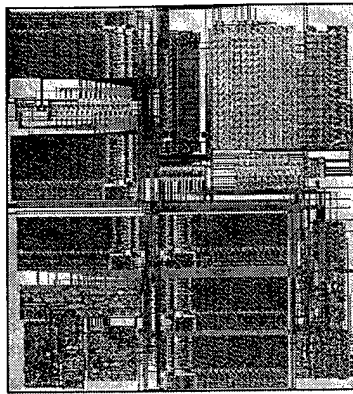
where $g(v)$ is the cost of an optimal path from S to v and $h'(v)$ is an estimate of the cost $h(x)$ of an optimal path from v to T . $h'(v)$ can be defined as the Manhattan distance between node v and the target, T .

3.6: Implementation

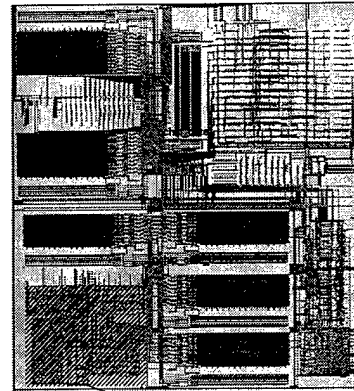
The framework for intrinsic area-array IC placement and routing has been implemented within the Berkeley *Magic* CAD Tool in C and run on a SUN SparcStation. Given the mask geometries of the chip core stored in CIF (Caltech Intermediate Form), our tool generates the area-array padframe of the IC. The pads are large areas of metal that are left unprotected by the overglass layer so they can be attached to bumps using the C4 process. Pad size is defined usually by the minimum size to which the solder bumps can be attached. The spacing of the pads is defined by the minimum pitch of the substrate. Both of these parameters are specified by the designer when running our tool. Before the tool proceeds to the pad placement, pad assignment, and pad routing steps, the chip core layout is flattened to extract the geometric information on the top few layers (the number of layers is specified by the designer), the pad keepout areas, the locations of I/O terminals, and the boundary of the core circuits. Utilizing this extracted information, the pad placement algorithm calculates the center of the chip core and then uses this center as a reference point and then generates a set of the potential area-array bonding pads on the top metal layer satisfying the user-specified bond pitch. Using a weighted bipartite matching algorithm, the I/O ports are assigned to area pads with the objective of minimizing the total wire length. This assignment generates a two-terminal netlist to be input into the pad routing routine to make the complete wiring. In the routing phase, the power/ground and clock I/O pads are routed first before the signal I/Os. The routing technique used is based on the general area routing algorithm [1, 8, 12]. The unrouted nets are routed manually with the computer-assisted interactive router [1]. The unused area pads located on the boundary of the padframe are designated as dummy pads. The final padframe (area-array pads along with the wire connections) is saved into a CIF file to be appended to the CIF file of the original design to form an area-array IC. In Figure 5, each of the substeps in the implementation of our tool is illustrated. The "Quantizer" chip (Figure 5(a)) was taken from the tutorial example of the EPOCH physical design tool and used as the implementation example.

After bringing up *Magic*, the first step is to import the layer information from the flattened CIF file into the internal layout representation in *Magic* with a modified built-in CIF reading routine. The I/O port locations are recorded for input into the pad assignment step. Figure 5(b) shows the layout of the top two metal layers extracted from the core design of the Quantizer.

The pad placement algorithm utilizes two of the database routines in *Magic* to place the pads on the top metal layer one at a time. In each step, the point-finding routine is used to locate the potential pad site. A searching box is created with its center corresponding to this pad site. Using the area enumeration routine, each of the tiles overlaps this searching box is determined to see whether any solid tile exists. If there is no solid tile for both the top-metal plane and the keepout-area plane, then an area pad is placed at this pad site. The placement is started from the center of the core, and grows outward to form an array of fixed-grid pads with spacing corresponding to the pad pitch. The placement process is terminated when the outmost row and column of the array are outside the boundary of the chip core and the number of pads is 10% greater than the number of I/O ports. The upper-right corner pad location is purposely omitted for chip orientation. After placing all the area pads, the keepout areas are cleared for routing. Figure 5(c) shows the result of

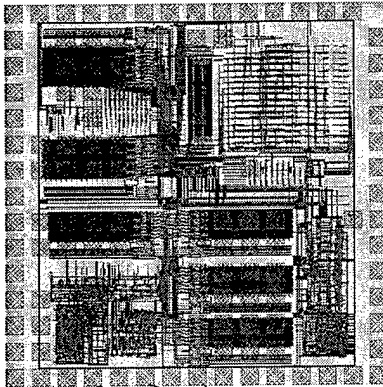


(a)



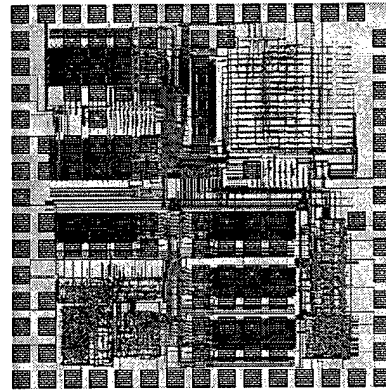
Pad keepout area highlighted

(b)

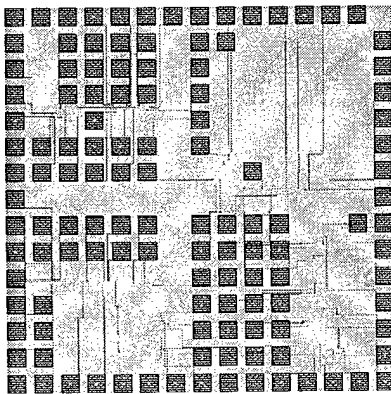


Area-array pads

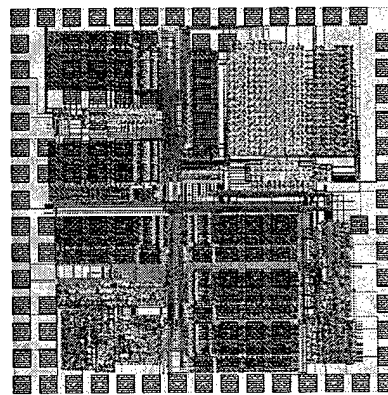
(c)



(d)



(e)



(f)

Figure 5. "Quantizer" (a) Core, (b) Core with Keepout Areas, (c) Area-Array Pads, (d) Routed Area-Array Pads, (e) Area-Array Frame and (f) Final Composite IC.

applying the pad placement algorithm. The pad size is $80\text{ }\mu\text{m}$ and the pitch is $120\text{ }\mu\text{m}$. The number of pads placed is 124 which is 10% greater than the number of I/O ports, 112.

After performing the pad assignment to generate a two-terminal netlist, net ordering is performed on the netlist to give the net closer to the center of chip core boundary a higher priority. Based on this ordering, at each net routing, we assign the I/O port as the source point and select the area pad as the destination area. Then *iroute* is called with these parameters to perform the routing. If the routing fails, we record this net into an array. If the routing of the net is successful, a glass cut and a label with suffix “_pad” are added to the targeted area pad. When all the nets have been processed, the unrouted nets are routed manually with computer assistance. Again, *iroute* can be used to find the location of the unrouted port and its assigned area pad by zooming into the area containing the I/O port and highlighting both the I/O port and the area pad. When all the nets are routed, the unused area pads located at the periphery of the pad array may be assigned as dummy pads. The remaining unused area pads are erased from the layout. Figure 5(d) shows the result of the routing on Quantizer chip.

When placement and routing of area-array pads are completed, our tool generates an area-array padframe by erasing all the layout information stored in the extracted CIF file and keeping intact the area-array pads and their routed wires to the I/O ports. This area-array padframe is then stored in a CIF file to be appended to the CIF file of original core design to form a complete area-array IC. Figure 5(e) shows the area-array padframe of the Quantizer chip and Figure 5(f) shows a complete area-array Quantizer IC. The chip has 112 bonding pads of size of $80\text{ }\mu\text{m}$ and pitch of $120\text{ }\mu\text{m}$. The total size of this version is $1.76 \times 1.76\text{ mm}$.

To further demonstrate our methodology of designing an intrinsic area-array IC, we conducted experiments on the design of three partitioned chips from the SUN MicroSparc design resulting from the study conducted in [4]. The SUN MicroSparc is a RISC processor design consisting of about 750,000 transistors. The design is described at the functional unit level consisting of the following: integer unit (IU), floating point unit (FU), memory management unit (MMU), data cache (D-CACHE), instruction cache (I-CACHE), s-bus controller (S-BUS-CTL), memory interface unit (MEM-INTF), clock control and buffers (CLK-CTL), and miscellaneous control logic (MISC). The partitioned resulted from [4] is summarized in Table 1.

Table 1. The Three Partitioned ICs of the SUN MicroSparc Processor.

Chip	# of I/Os	Modules
A	485	D-CACHE, I-CACHE, MMU
B	298	FU, IU
C	414	MEM-INTF, SBC, CLK-CTL, MISC

All three chips were resynthesized and laid out using the EPOCH physical design automation tool. The core circuits were exported to CIF files. The size of the core circuits for ChipA, ChipB, and ChipC were $5957.5 \times 5806.6 = 3.46e + 07\text{ }\mu\text{m}^2$, $6809.6 \times 6508.2 = 4.43e + 07\text{ }\mu\text{m}^2$, and $1500.8 \times 2119.2 = 3.18e + 06\text{ }\mu\text{m}^2$, respectively. In this paper, we present the results for ChipA and ChipB only. Figure 6(a) and Figure 6(b) show the area-array padframe and the complete area-array IC of chipA, respectively. The pad size is $100\text{ }\mu\text{m}$ and the pitch is $200\text{ }\mu\text{m}$. Metal-3 is the top metal. The area pads were not placed

over the area of RAM modules of the chip. The pad routing was done on the metal-2 and metal-3 layers. The routing of the area pads took approximately 8 hours on an UltraSparc 170 with 512 Megabytes of internal memory. There were 46 nets which were not routed automatically. It took another hour or so to manually route these nets using *iroute* with the assistance provided in our method. The reason it took so long to route the I/O ports to their assigned area pads was due to the clustering of the I/O port locations at certain regions in the core area. In this case most of the I/O ports were located at the bottom-left of the core area. Hence, some of the I/Os were assigned pads that were located at the top half of the core area. Therefore, the routing of these nets consumed a lot of time and memory space. If we could scatter the I/O ports all over the core area, the routing would be faster. The total area of this chip is $6700 \times 7500 = 5.025e + 7 \mu m^2$.

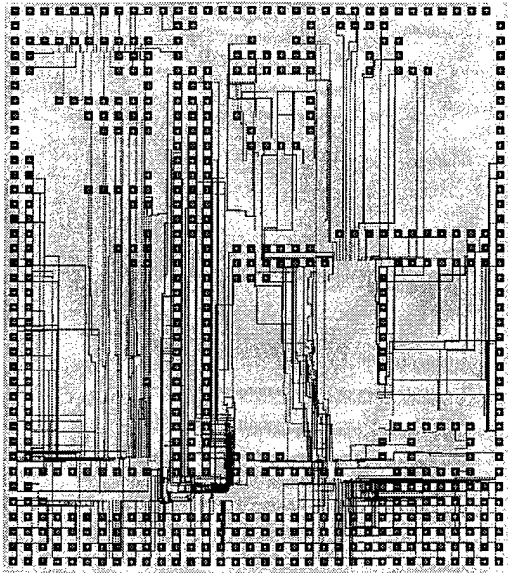
The next design was ChipB of the MicroSparc RISC processor. Figure 6(c) and Figure 6(d) show the area-array padframe and the complete area-array IC of ChipB, respectively. The pad size is $100 \mu m$ and the pitch is $225 \mu m$. Metal-3 is the top metal. The pad routing was done on the metal-2 and metal-3 layers. The routing of the area pads took approximately 6 hours on an UltraSparc machine for the same reason explained in the ChipA case. There were 16 nets which were not routed automatically. The total area of this chip is $7300 \times 6850 = 5.0e + 7 \mu m^2$.

4: Summary & Conclusions

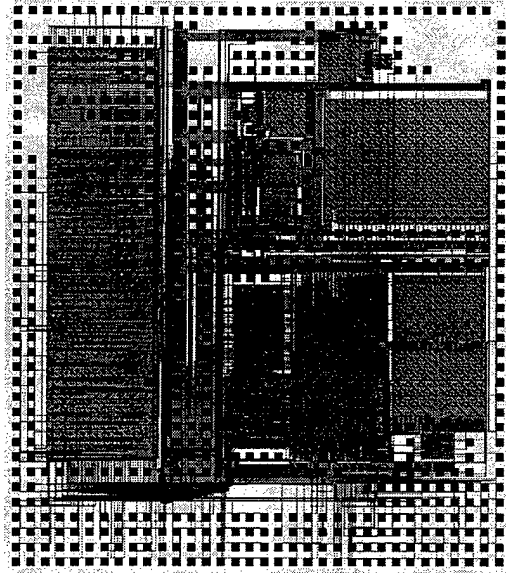
The framework for designing an intrinsic area-array IC has been developed in this paper. This framework is divided into several substeps: design guideline definition, data preparation, pad placement, pad assignment, pad routing, and output padframe generation. For each of these substeps, the solution method and its implementation have been presented. Though further work is necessary, the results of the current experiments with several design examples are promising and satisfactory.

The method developed in this paper has several advantages over some previous approaches: there is no need for extra layers for placing the area-array pads and redistribution. In fact, the placement and routing of area pads are performed on the pre-existing metal layers. The resulting area-array IC should be smaller in size and cheaper in cost for high I/O count. Our method can also serve as a postprocessor to any IC physical design tool for designing area-array ICs. In our implementation, we used the EPOCH physical design automation system developed by Cascade Design Automation to synthesize and layout the core circuits of the design following the guidelines provided in our method. The layout was then exported to a CIF file to be imported into our tool. An area-array padframe was generated to be appended to the core design to form an intrinsic area-array IC. In generating the padframe for the Quantizer chip, it took approximately 6 minutes to place and route 112 area pads. The success rate of the routing was close to 99.0% so only 1% of the nets needed to be manually routed. Even for a large design with close to 500 I/Os (ChipA), it took less than 6 hours to finish placement and routing of the area pads with 46 unrouted nets. It took another hour or so to manually route these nets using *iroute* with the assistance provided in our method. Our method can also be incorporated into any physical design CAD tool to provide area-array pad placement and routing capability.

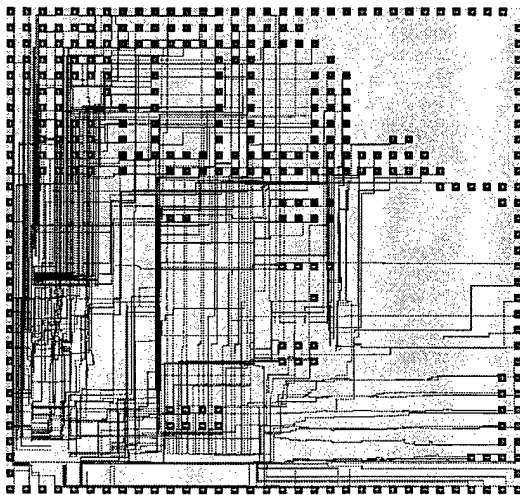
Placing and routing area-array pads by hand will be virtually impossible and very time-consuming as the number of I/Os becomes large. As flip-chip design approaches with high I/O count become more common in the future, there will be a greater demand to place and route area-array pads for hundreds of I/Os automatically. The framework presented in this paper for designing an intrinsic area-array IC will be useful in this case.



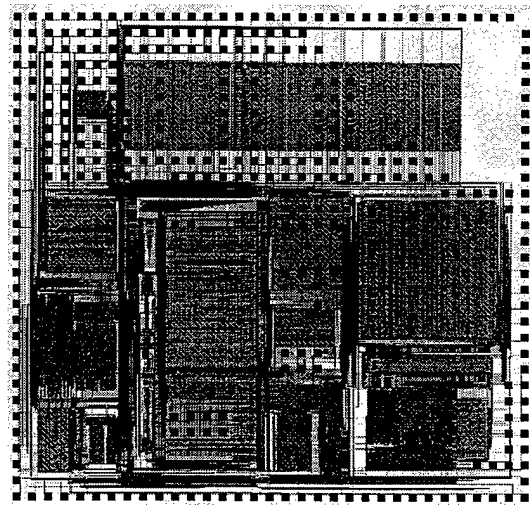
(a)



(b)



(c)



(d)

Figure 6. Results: (a) Area-Array Padframe for ChipA, (b) Final Composite ChipA, (c) Area-Array Padframe for ChipB, and (d) Final Composite ChipB.

Acknowledgement

The Epoch Physical Design tool by Cascade Design Automation and the MCM Station software by Mentor Graphics have been extensively used for this research. The authors gratefully acknowledge the donation of these two tools to the University of Tennessee. Last but not least, the authors also acknowledge the contribution of the *Magic* layout tool to this work.

References

- [1] M. H. Arnold and W. S. Scott. An Interactive Maze Router with Hints. In *Proc. 25th Design Automation Conf.*, pages 672–676, Anaheim, CA, June 1988.
- [2] J. P. Cohoon and P. L. Heck. BEAVER: A Computational-Geometry-Based Tool for Switchbox Routing. *IEEE Trans. on Computer-Aided Design*, CAD-7(6):684–697, 1988.
- [3] J. Darnauer and W. W. Dai. Fast Pad Redistribution for Periphery-IO to Area-IO. In *Proc. Multichip Module Conf.*, pages 38–43, Santa Cruz, CA, March 1994.
- [4] P. Dehkordi, K. Ramamurthi, D. Bouldin, H. Davidson, and P. Sandborn. Impact of Packaging Technology on System Partitioning: A Case Study. In *Proc. Multichip Module Conf.*, pages 144–149, Santa Cruz, CA, 1995.
- [5] P. Dehkordi, C. Tan, and D. Bouldin. Intrinsic Area-Array ICs: What, Why, and How. In *Proc. Multichip Module Conf.*, pages 120–124, Santa Cruz, CA, February 1997.
- [6] J. Dion and L. M. Monier. Contour: A Tile-based Gridless Routing. Technical Report WRL Research Report 95/3, Digital Equipment Western Research Laboratory, March, 1995.
- [7] R. Farbarik, X. Liu, M. Rossman, P. Parakh, T. Basso, and R. Brown. CAD Tools for Area-Distributed I/O Pad Packaging. In *Proc. Multichip Module Conf.*, pages 125–129, Santa Cruz, CA, February 1997.
- [8] Gordon Taro Hamachi. An Obstacle-Avoiding Router for Custom VLSI. Technical Report UCB/CSD 86/291, Univ. of California, Berkeley, CA, April 1986.
- [9] R. Jonker and A. Volgenant. A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems". *Computing*, 38:325–340, 1987.
- [10] F. E. Kiamilev, A. V. Krishnamoorthy, J. Rieve R. G. Rozier, G. F. Aplin, C. D. Hull, R. Farbarik, and R. E. Oettel. Design of ICs for Flip-Chip Integration with Optoelectronic Device Arrays. In *Proc. Multichip Module Conf.*, pages 163–167, Santa Cruz, CA, February 1997.
- [11] G. Kromann, D. Gerke, and W. Huang. A Hi-Density C4/CBGA Interconnect Technology for a CMOS Microprocessor. In *Proc. IEEE 44th Electronic Components & Technology Conf.*, pages 22–28, Washington, D. C., May 1994.
- [12] C. Y. Lee. An Algorithm for Path Connections and Its Applications. *IRE Trans. Electronic Computers*, pages 246–265, September 1961.
- [13] T. Ohtsuki. *Layout Design and Verification*, volume 4 of *Advances in CAD for VLSI*, chapter Maze-Running and Line-Search Algorithms. Elsevier Science Publishing Company, North-Holland, Amsterdam, 1986.
- [14] J. K. Ousterhout. Corner Stitching: A Data Structuring Technique for VLSI Layout Tools. *IEEE Trans. on Computer-Aided Design*, CAD-3(1):87–100, 1984.
- [15] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, Englewood Cliffs, New Jersey, 1982.
- [16] H. Shin and A. Sangiovanni-Vincentelli. Mighty: A Detailed Router Based on Incremental Routing Modifications. *IEEE Trans. on Computer-Aided Design*, CAD-6(6):942–955, November 1987.
- [17] J. Soukup. Circuit Layout. In *Proc. of the IEEE*, pages 1281–1304. IEEE, October 1981.
- [18] R. R. Tummala and E. J. Rymaszewski, editors. *Microelectronic Packaging Handbook*. Van Nostrand Reinhold, 1989.

York, J., Powell, T., Dehkordi, P. and D. Bouldin,
“Enhancement of MCM Testability Using an Embedded Reconfigurable FPGA” ,
Proceedings of 1997 International Conference on Innovative Systems in Silicon, pp. 165-173, Austin,
TX, October 8-10, 1997.

Enhancement of MCM Testability Using an Embedded Reconfigurable FPGA

John York, Tim Powell, Peyman Dehkordi and Don Bouldin*

Department of Electrical Engineering

The University of Tennessee

Knoxville, TN 37996-2100

E-mail: york@microsys1.engr.utk.edu

Abstract

The testability of an MCM can be enhanced significantly for very little cost whenever a reprogrammable FPGA component that is already embedded in the MCM for functionality is utilized for diagnostics. This approach can have some of the characteristics of a smart substrate which uses the scan cell beside-the-signal-path (BSP) methodology. The design and implementation of an MCM with this capability is presented along with descriptions of the self-test algorithms, fault isolation and real-time testing and monitoring that this method provides.

1. Introduction

One of the major obstacles inhibiting MCMs to be a widely accepted and mass-produced technology is testability. In most cases, MCM testing involves the verification of substrate interconnects, logical connections and IC interaction functionality. Testing methods for ICs are difficult to apply to MCMs since they often contain a heterogeneous mix of components. The testing of MCMs can involve the use of capacitive or resistive probing with flying probes, high-density probe cards or electron-beam probe testing, all of which can be expensive and time consuming. It is possible for the cost of testing to exceed the design and fabrication cost of the MCMs themselves [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].

One method of reducing test cost and time, thereby reducing overall MCM cost, is the utilization of boundary scan features found on some ICs devices. If all devices on the MCM have boundary scan features, a complete substrate test is possible. Unfortunately, most IC devices, including custom designed chips, do not incorporate boundary scan circuits, so other methods of testing are needed [4].

Known good die (KGD) is an issue for MCM testability, since previously untested dies may be assembled in a MCM system. With the advent of different levels of KGD, the functionality of ICs in a MCM can be assured *at some level* depending on cost. However, the IC interconnects and system functionality must still be tested as a complete module. There are two drawbacks with depending on KGD for a MCM design. First, fully tested KGD of the highest level can be

*The authors gratefully acknowledge the support of DARPA grant DAAH04-94-G0004.

MCM MODULE

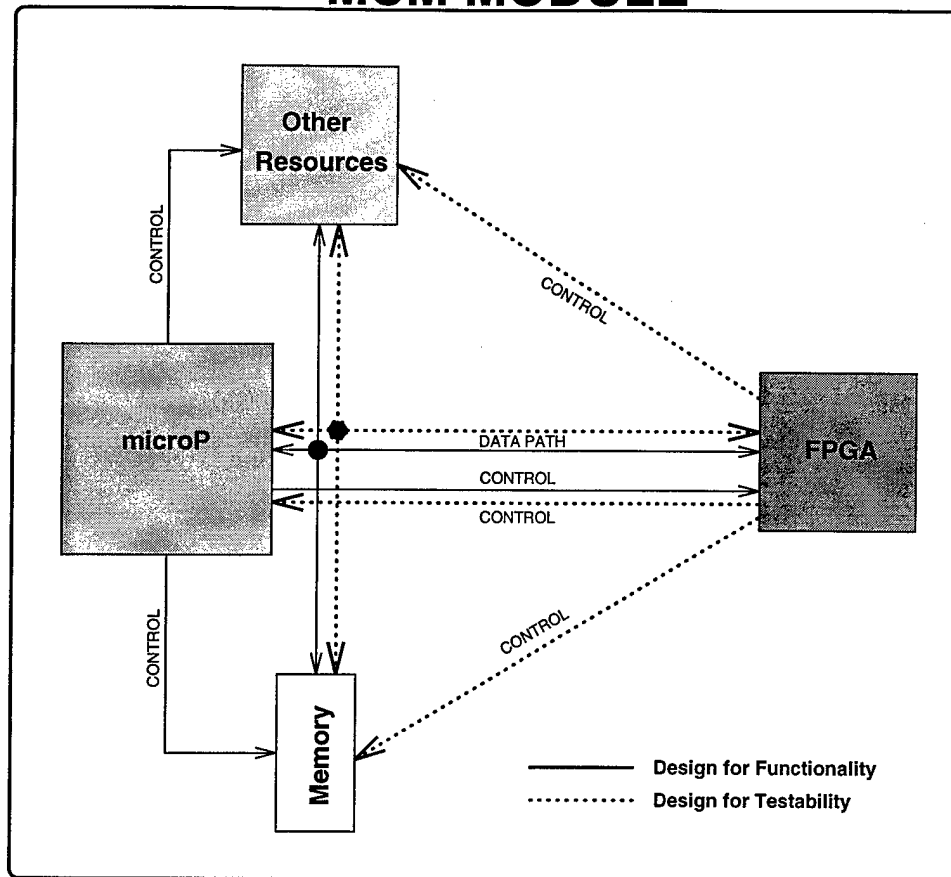


Figure 1: Designing for Testability for a Typical MCM.

relatively expensive compared to their IC packaged equivalents. This can substantially raise the cost of a MCM, thereby overwhelming the benefits of using thoroughly tested MCMs over an equivalent printed circuit board (PCB). Second, not all ICs on the market today are available as KGD at any level.

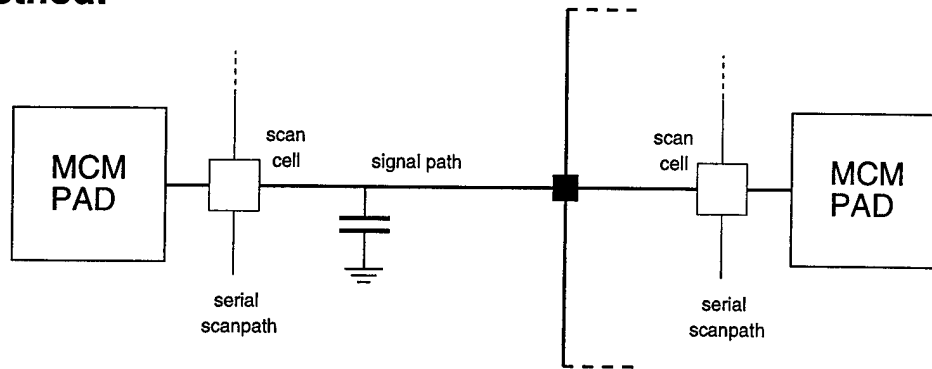
Section 2 of this report will discuss some of the testability features an embedded reconfigurable FPGA can provide and will very briefly describe the concept of BSP Smart Substrates. In Section 3, the design and implementation of a prototype MCM will be given to illustrate the versatility of a reconfigurable FPGA in enhancing testing. Conclusions are presented in Section 4.

2. Testing with an Embedded FPGA

MCMs with an embedded reconfigurable FPGA in the design can enhance its testability. As with some IC designs, the MCM system must be designed and planned for testability to maximize the effectiveness of the testing. As shown in Figure 1, the MCM design should not only consider functionality but also testability.

Critical signal paths can be connected to the FPGA I/O, and some of the FPGA logic can be configured as scan cells for boundary scan. This setup is similar to that of the BSP method for

ISP Method:



BSP Method:

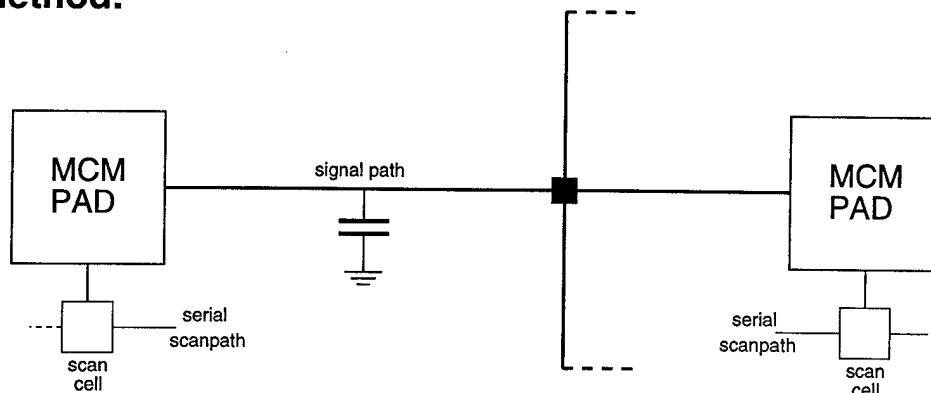


Figure 2: Placement of the Scan Cells In and Beside the Signal Path [4].

Smart Substrates. Smart Substrates have the potential to tolerate incompletely tested dies and can provide the means to control and monitor individual MCM dies from the module's pins. In the BSP Smart Substrate method, scan cells within the substrate are placed beside the signal path to be tested as shown in Figure 2. The figure also shows the ISP (In Signal Path) method scan cells, which has similar benefits as the BSP method except the scan cells are placed in series along signal paths. Since the BSP method places scan cells in 'parallel' to the MCM pads, only a small capacitive load is seen during regular operation when the scan cells are disabled. The BSP method can provide the features of standard IEEE 1149.1 boundary scan, and can allow for bare substrate tests [3, 4]. However, BSP testing using an embedded FPGA would be possible only after final assembly of the dies onto the substrate so bare substrate tests would be impractical. Fortunately, the I/Os on some FPGAs can be tri-stated to reduce capacitive effects on signal lines.

Careful planning for testability, proper connectivity and configuration of the FPGA not only can allow for an emulated BSP methodology, but can also allow thorough testing of the MCM in its environment. The FPGA can be configured as a pseudo built-in logic analyzer that can monitor core dies within the MCM in real time. This allows testing of not only the hardware aspect, but also allows monitoring of software execution in a limited manner. Moreover, a FPGA can be configured as a self-tester for the module by creating stimuli and monitoring results. If the FPGA has access to the external I/O of the module itself, testing results can be sent as output and observed as they become available. With the proper reconfigurable resources and

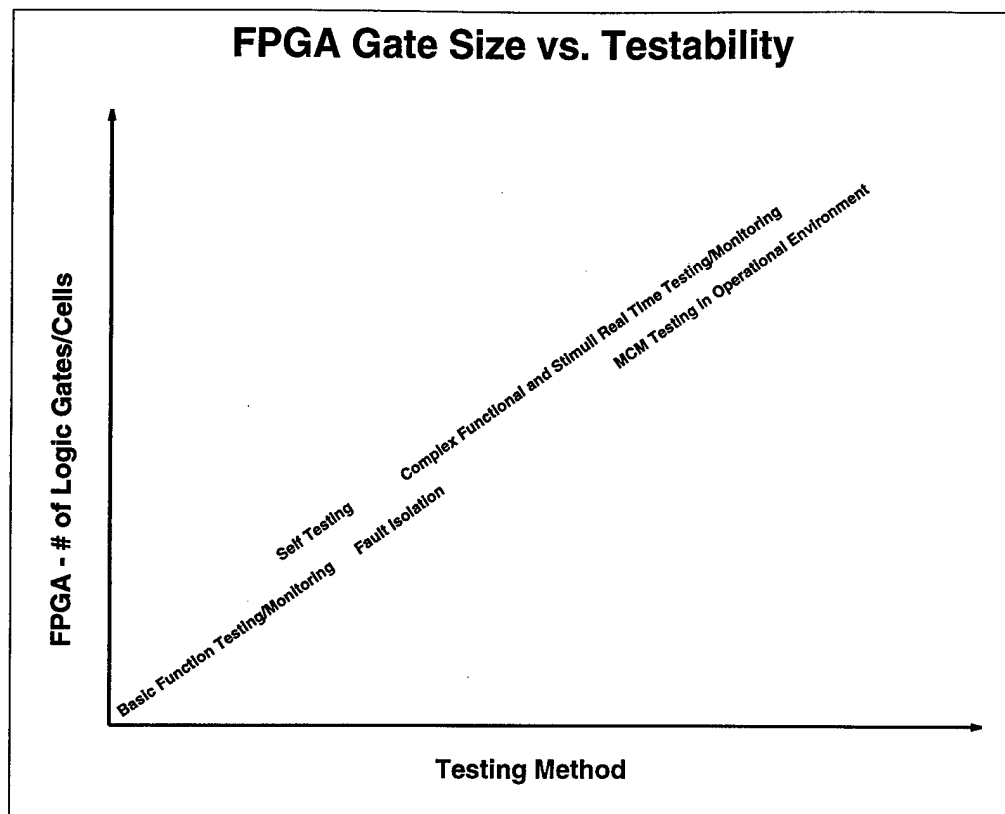


Figure 3: FPGA Gate Size vs Complexity of Testing.

connectivity, the FPGA can provide all of the above forms of testing in a virtually self-contained unit.

It has been stated above that the testability of a MCM can be enhanced with an embedded FPGA with the proper connectivity and configuration. The question arises as to what size (number of logic gates/cell blocks) FPGA to use and what connections must be made between the FPGA and other ICs on the MCM and between the FPGA and MCM I/O. The optimum size is dependent on the particular design and is currently under investigation for our specific example MCM described in the next section. The questions are important since a FPGA can occupy a significant amount a routing and substrate area. The size of the FPGA can depend on the complexity of the MCM and the desired complexity of testing required. Figure 3 gives a qualitative illustration of the spectrum of possibilities.

It can be assumed that there is no one setup that will work for all MCMs, as applications of MCMs vary. However, for processor-based MCMs, some common bases can be set. In a processor-based MCM, the critical signal paths will likely be the address and data buses. Therefore, the FPGA I/O should tap into those signal lines. In order for the FPGA to isolate chips for testing or fault identification, the FPGA needs access to the control lines of the other ICs. This allows the FPGA to turn off chips for isolated testing and enables an algorithm to identify a functionally faulty chip in the system. As with the BSP Smart Substrate method, the necessity of using KGD can be avoided. With proper rework facilities, faulty chips could be replaced on the MCM to increase yield. While other testing methods such as the BSP Smart Substrate method may tolerate the use of all untested dies, a FPGA used for testing would

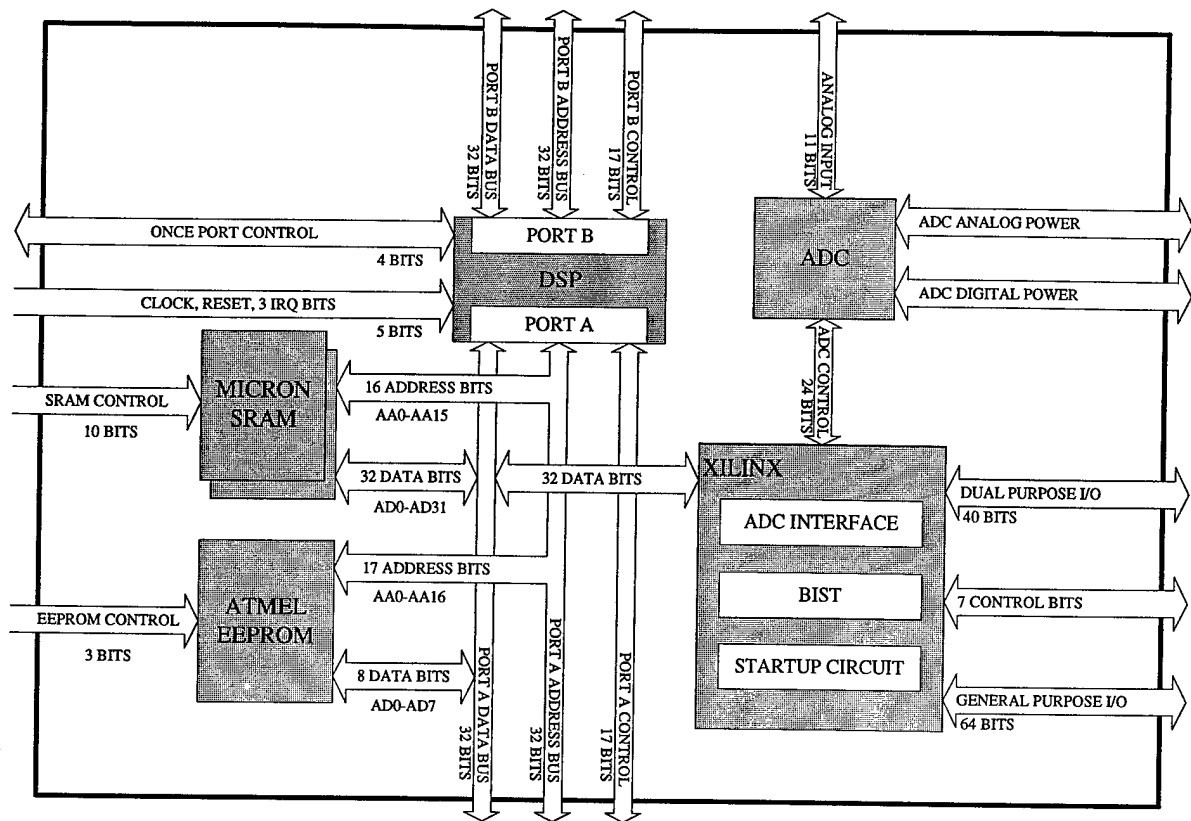


Figure 4: Block (Flow) Diagram of DSP MCM.

obviously have to be a KGD, preferably of the highest level.

In general, it would not be a wise decision to include a FPGA in a MCM design just for testing purposes since FPGAs can take up a significant amount of area on the MCM substrate. It is recommended that an embedded FPGA be used for testing purposes if a MCM core design already incorporates a reconfigurable FPGA in the unit. Unfortunately, design constraints may not allow an embedded FPGA to be used for testing. For instance, to perform testing of a MCM may require the FPGA to be larger in gate size, have more I/O or require additional routing within the substrate. The gate size of the FPGA can be increased without significantly increasing the overall size of the IC. However, even with the additional gates, using the FPGA for testing may still be impractical if additional routing is required and routing within the substrate is saturated. If a FPGA is included in a MCM design for resource or glue logic, and the design is adequately flexible to allow for additional routing and/or increases in FPGA gate and/or I/O size, design considerations in I/O routing must be made so that the FPGA can be established as a KGD. It is essential that the FPGA be tested to be a KGD before it is used to test the remainder of the MCM system. If the FPGA is not established as a high level KGD, it could be difficult to determine sources of random occurring faults. The ideal case would be to manufacture MCMs with high level, KGD FPGAs.

3. Prototype MCM

Figure 4 shows a block diagram of an MCM designed at the University of Tennessee [11].

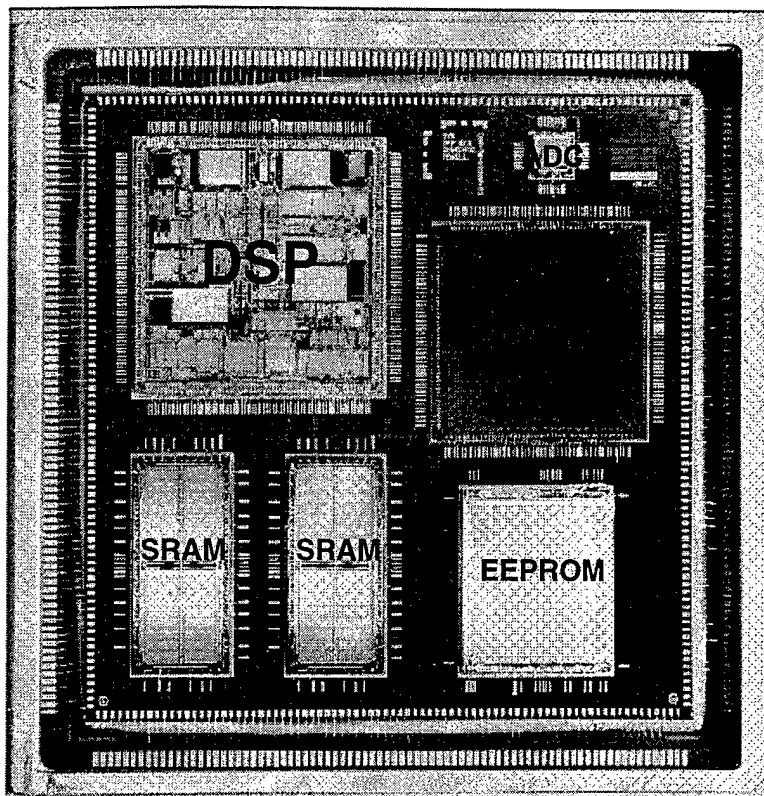


Figure 5: Photograph of the DSP MCM.

There are a total of six ICs on the MCM including a Motorola 32-bit digital signal processor (DSP). Figure 5 shows a photograph of the MCM which was fabricated by Micro Module Systems via the MIDAS Service. The other ICs include two Micron 64K x 16 SRAM, an Atmel 128K x 8 EEPROM, a Xilinx XC4010 series FPGA and a 12-bit Analog-to-Digital Converter (ADC) from the nearby Oak Ridge National Laboratory. Note that a full data bus (Port A) is interconnected to the Xilinx 4010 FPGA die within the module. Since this was an experimental prototype and none of the dies were KGD, most of the I/O of the FPGA (and other ICs) were brought out as external I/O for the module. This helps isolate ICs for testing and fault isolation. The MCM I/O of the FPGA are 'wrapped' back into the MCM for connection to the address bus (Port A) and control bus of the DSP processor and other ICs in the module. Some of the FPGA-MCM I/O can be used to communicate with the MCM's environment. A PCB is currently being developed to allow for these 'wrapped' back connections, to provide resources for FPGA configuration and to allow communication between the MCM and its environment. Figure 6 shows a block diagram of the PCB that will be used to provide this environment.

It is important to note that the FPGA in the MCM was not originally designed into the system for testability. The FPGA was deemed as necessary glue logic to provide operational resources for the system. For example, the FPGA provides logic for an address decoder, an interactive user I/O interface and the means for reconfigurable computing with the DSP processor. During the development process, concern for testing of the MCM became an issue. It was decided that with the flexibility of the MCM I/O and the proper environment, the FPGA could be used to test the MCM system.

Note from the PCB block diagram that auxiliary resources, such as an address decoder, are provided for the MCM. This is to help establish the FPGA as a KGD before it is used for testing,

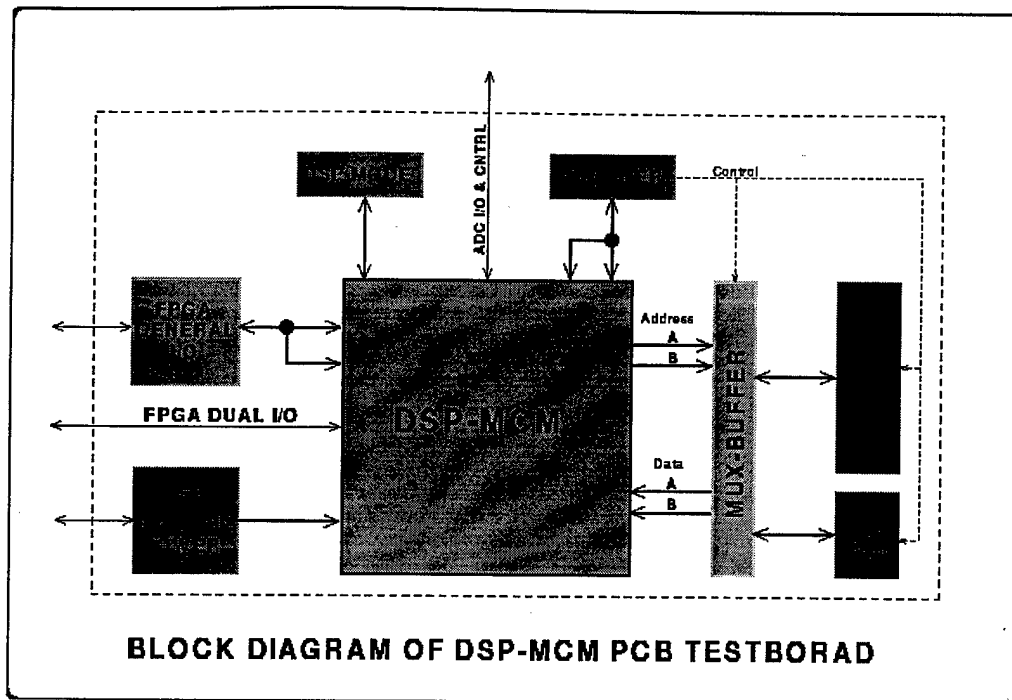


Figure 6: Block (Flow) Diagram of the Test Environment PCB for the DSP MCM.

and allow testing of the MCM to be continued regardless of certain faults that may occur within the module. Also notice from the diagram that there are 'wrapped around' connections on the PCB for the MCM. The 'wrapped around' routing, as described before, allows connections between the FPGA and the main address bus and the control lines of the other ICs, including the main processor unit. The PCB is designed to allow the FPGA to be the main interface between the MCM and its environment and will be used to establish the FPGA as a KGD at the functional, at-speed level.

Figure 7 shows the actual layout of the MCM testboard PCB. Note the many intervias between the MCM and PCB resources. The board is comprised of four signal layers and two (VCC and GND) power layers. This level of complexity is needed to provide the MCM a testing and functional environment. If the MCM were to be manufactured as a stand alone application unit, many of the 'wrapped around' connections (MCM to glue logic resources to MCM) would be fabricated with the substrate of the MCM. For example, on the PCB, address lines from the prototype MCM are traced to a PAL device, and then traced back to the MCM. This was necessary since an address decoder was not incorporated onto the MCM. The PAL would be configured as a custom address decoder. For an application, the address lines would be routed internally to the embedded FPGA, which would be configured as an address decoder, among other things. However, since this MCM was a prototype, most I/O lines within the MCM were routed to external pins to aid in testing of the MCM, and help establish ICs within the MCM as functional, partially functional or nonfunctional.

Once the FPGA is established as a functional unit (KGD), testing of the MCM and its components can begin. One of the simplest testing that can be done is data monitoring (pseudo logic analyzer) with fault isolation. For example, the FPGA can be configured to 'turn off' all of the ICs except for the internal EEPROM and DSP processor. The Motorola DSP processor has a built-in boot algorithm that allows the loading of the 4K instruction set from a byte-wide memory source. Simple instruction codes can be loaded into the EEPROM. With the DSP initiated in

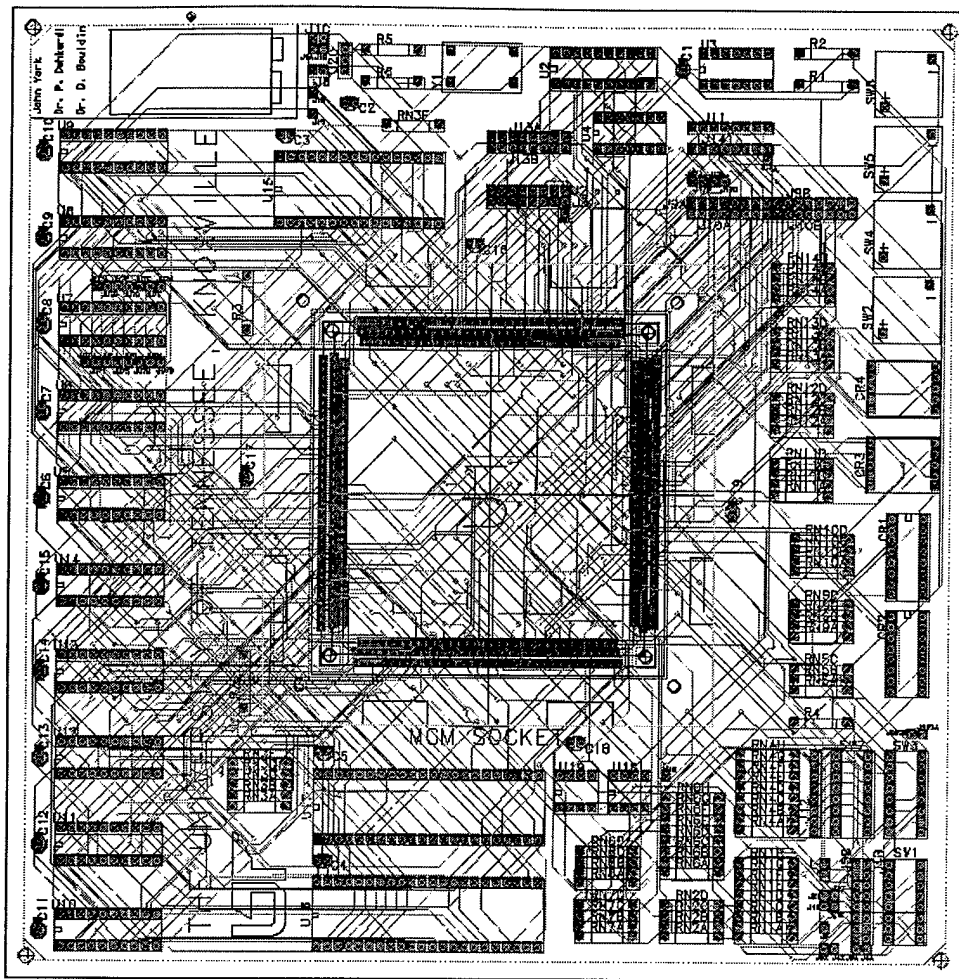


Figure 7: Actual Layout (Place and Route) of the PCB in Layout Tool (Mentor Graphics)

the proper mode, transfer of address code and data (instruction code) can be monitored by the FPGA. Since the address bus has to go through the FPGA first for decoding, it can determine if the DSP processor is sending the proper addresses. This should be simple since the addressing will be sequential and an initial address value will be known.

Monitoring of the address bus will validate or invalidate not only the DSP's address capability, but also the routing lines from the DSP's address bus. If the instruction set within the EEPROM is kept to an alternating, repeating code, the FPGA can monitor the data bus for errors from the EEPROM. Since this particular FPGA has an internal system clock, it can stop operations once an error or fault is detected and notify its environment (PCB - LEDs) the probable cause of the error/fault. If the FPGA receives an incorrect address midway through the boot cycle, it can flag the DSP as the fault cause. Then, if an incorrect instruction code is seen, it can flag the EEPROM as the cause of the error. However, if incorrect address values and/or incorrect instruction codes are seen by the FPGA continuously from the start, it will be difficult to determine whether the device or the routing are at fault. In this case, a more complex algorithm will be needed for the FPGA to determine the difference between routing faults and device faults.

4. Conclusions

MCM designs that include a reprogrammable FPGA for glue logic can have the benefits of enhanced testability. Even though additional design costs may be needed to consider using the FPGA for testing purposes, the benefits of testability can far outweigh those costs, depending on the interconnections and size of the FPGA in the design. Again, it is not recommended that an FPGA be incorporated into an MCM solely for testing. Design for testability utilizing an embedded FPGA is recommended only when an FPGA is already needed as a resource for the MCM system.

5. References

- [1] Andrew Flint and James Trent, "MCM Test Techniques: Keeping Pace With Packaging Technology", in *Electronic Packaging & Production*, January 1996, pp. 49-52.
- [2] Andrew Flint and William Blood Jr., "MCM Test Strategy: Board Test in an IC Environment", in *ICEMM Proceedings'93*, 1993, pp. 429-434.
- [3] A. E. Gattiker, W. Maly, and M. E. Thomas, "Are There Any Alternatives to 'Known Good Die' ?", in *Proceedings of 1994 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, 1994, pp. 102-107.
- [4] H. Werkmann, B. Laquai, and T. Schwederski, "Efficient Smart Substrates with Test Capabilities and On-Line Temperature Monitoring", in *Proceedings of 1995 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, 1995, pp. 183-188.
- [5] Ken Posse, "Algorithmic Diagnosis of Multichip Module Defects Using the IEEE 1149.1 Standard", in *Proceedings of 1994 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, 1994, pp. 199-204.
- [6] Lynn E. Roszel, "MCM Test Methodologies and Project Experiences", in *ICEMM Proceedings'95*, 1995, pp. 424-427.
- [7] M. J. Begay, J. Van Zee, G. Westbrook, and B. Williams, "Getting to Know Your MCM Die", in *ICEMM Proceedings'93*, 1993, pp. 160-165.
- [8] Stuart A. Berke, David J. Golden, and Edwin A. Rogers, "Test Strategy for a Microprocessor Based Multi-Chip Module", in *ICEMM Proceedings'95*, 1995, pp. 419-423.
- [9] Stephen C. Hilla, "Boundary Scan Techniques in an MCM-D Application", in *Proceedings of 1994 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, 1994, pp. 217-222.
- [10] William Blood Jr. and Andrew Flint, "Design and Test of a Complex MCM Product", in *Proceedings of 1994 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, 1994, pp. 193-198.
- [11] P. Dehkordi, T. Powell, and D. Bouldin, "Development of a DSP/MCM Subsystem Assessing Low-volume, Low-cost MCM Prototyping for Universities", in *Proceedings of 1996 IEEE Multi-Chip Module Conference*, Santa Cruz, CA, February 6-7 1996, pp. 360-370.